

**Приложение 4.**  
**Фрагмент конспекта лекций по программе "Работа с  
информационными и платежными системами, построенными на базе  
использования *Internet*-технологий"**

**Общие сведения об Интернет**

Интернет (Internet) – это сеть, объединяющая отдельные локальные, региональные, национальные и глобальные сети, это глобальное сообщество мировых ИВС, которые используют для информационного обмена семейство протоколов TCP/IP.

Дословно термин «Internet» означает «между сетей». Это отражает основную функцию Internet - объединение не только отдельных ЭВМ (хост-машин), но и обеспечение связи между различными сетями в глобальном масштабе. Это объединение даёт возможность обмена информацией между всеми ЭВМ, входящими в сети, подключённые к Internet. При этом не важно, в какой операционной системе работают хост-машины (Windows, UNIX и т.п.).

Справка. Определение термина Internet было дано 24.10.1995 года Федеральным советом США по компьютерным сетям (FNC - Federal Networking Council): "Internet - это глобальная информационная система, которая логически соединена посредством адресного пространства, основанного на протоколе IP (Internet Protokol) или заменяющих его протоколов, способна поддерживать передачу данных посредством протокола TCP (Transmission Control Protocol) или заменяющих его протоколов, обеспечивает, использует или делает доступными услуги по передаче данных и соответствующую инфраструктуру".  
История создания сети Интернет

1962 г.: Пол Бэран из американского мозгового центра времен «холодной войны» (Rand Corporation) предложил коммутацию пакетов (КП) в качестве надежной сетевой технологии.

Леонард Клейнрок разработал базовые принципы пакетной коммутации, ставшие основой Internet.

1964 г.: Rand Corp. публикует концептуальные положения будущей надежной сети ПД.

1969 г.: Агентство перспективных исследований Министерства обороны США (DARPA) финансирует проект создания сетей с КП и принимает решение объединить суперкомпьютеры оборонных, научных и управляющих центров в единую сеть, которая получила название ARPANET (Advanced Research Projects Agency Network). Первая действующая информационная сеть ARPANET, объединила компьютерные системы университетов Лос Анджелеса, Стэнфорда, Санта Барбары и Солт Лейк Сити.

1969 г. – 4 ГВМ; 1972 г. – 14; 1974 г. – 37; 1979 г. – 111 ГВМ и ТМ. Через 25 лет (1994 г.) – около 4 млн. ГВМ и ТМ.

**Характеристика сети ARPANET**

L пакета (максимальная) = 1008 бит;

L пакета (средняя) = 218 бит;

50% пакетов являются служебными – команды и ответы;

Повышение достоверности – механизм квитирования и тайм-аутов (» 250 мс), ПОС;

Надежность: в среднем 1 ошибка на 12880 переданных пакетов;

Кгот = 0,9836;

Маршрутизация – табличная, через каждые 640 мс каждый узел формирует маршрутные сообщения, направляемые всем смежным узлам, с информацией о задержках.

На основании этих данных узлы выбирают направления передачи, которые соответствуют минимальной задержке.

### Появление Интернет

1. 1974 г.: Вinton Серф и Роберт Канн публикуют основные принципы работы протоколов TCP/IP.
2. 1980 г.: История глобальной сети Интернет начинается примерно с 1980г., когда ARPA стало переводить компьютеры, подключенные к своим исследовательским центрам, на протоколы TCP/IP. Модернизацию начали с ARPANET.
3. 1982 г.: Для сети ARPANET утверждено семейство протоколов TCP/IP.
4. 1983 г.: Штаб-квартира Минобороны США объявила, что все их компьютеры переведены на TCP/IP. В этом же году Минобороны США разделило ARPANET на две независимые сети: научно-исследовательскую – ARPANET и военную – MILNET.
5. 1984 г.: Национальный научный фонд США (NSF) начал инвестировать научную компьютерную сеть NSFNET.
6. 1986 г.: Создание Национальным научным фондом США компьютерной сети NSFNET, которая объединила научные центры и университеты США. В качестве базовых протоколов были выбраны протоколы TCP/IP. К NSFNET примкнули NASA, DOE (Министерство энергетики), DOD (Министерство обороны) и Национальный институт здравоохранения. Появились шесть первых имен доменов: gov, mil, edu, com, org и net.
- 1986 г. можно считать годом становления глобальной компьютерной сети Интернет с опорной сетью NSFNET.
7. 1989 г.: Последний год ARPANET (руководство ARPANET не сочло возможным войти в проект NSFNET и дальнейшее развитие Интернет (Internet) продолжалось уже без ARPANET).
8. Конец 1995 г.: 6,6 млн. ГВМ; 70 тыс. независимых сетей и 200 тыс. сегментов. Динамичное развитие спектра услуг пользователей Интернет. IS-провайдеры. В 1997 г. – около 3500 тыс. провайдеров, предоставляющих услуги сети Интернет.

Сейчас Internet составляют более 200 тыс. отдельных сетей, связывая более 2 млн. узловых компьютеров в 150 странах мира. Более 350 млн. пользователей регулярно используют ресурсы Internet.

Сама сеть Internet не имеет владельца, однако она соединяет множество сетей ЭВМ, которые имеют своих владельцев. Многие из таких сетей ЭВМ (либо отдельные хост-ЭВМ) предоставляют на коммерческой основе различную информацию, полезную во многих сферах жизнедеятельности человека. Эта информация накапливается в информационных банках национальных сетей, а доступ обеспечивается средствами Internet, что, собственно, и объясняет всемирную популярность Internet.

У истоков Internet в России стоят компьютерные сети ОИЯИ (г. Дубна) и Института им. Курчатова И. В. (г. Москва).

Высшая школа – естественный и активный участник работ по развитию Internet в России. В 1993 году в Госкомвузе РФ были разработаны концепция и программа создания российской университетской компьютерной сети, которая получила название RUNNet (Russian UNiversity Network). Сеть RUNNet необходима для достижения двух целей: формирования единого информационного пространства российской высшей школы и его интеграции в мировую информационную систему образования, науки и культуры, развивающуюся в рамках глобальной сети Internet. В 1994-95 годах была создана основа RUNNet – опорная сеть, обеспечивающая магистральную связь между всеми экономическими регионами России и подключение к Internet через зарубежные академические сети. В эту опорную сеть включены компьютерные узлы крупных научных и учебных центров страны, связанные между собой спутниковыми каналами.

# Лекция 1. Разновидности сайтов в Интернет

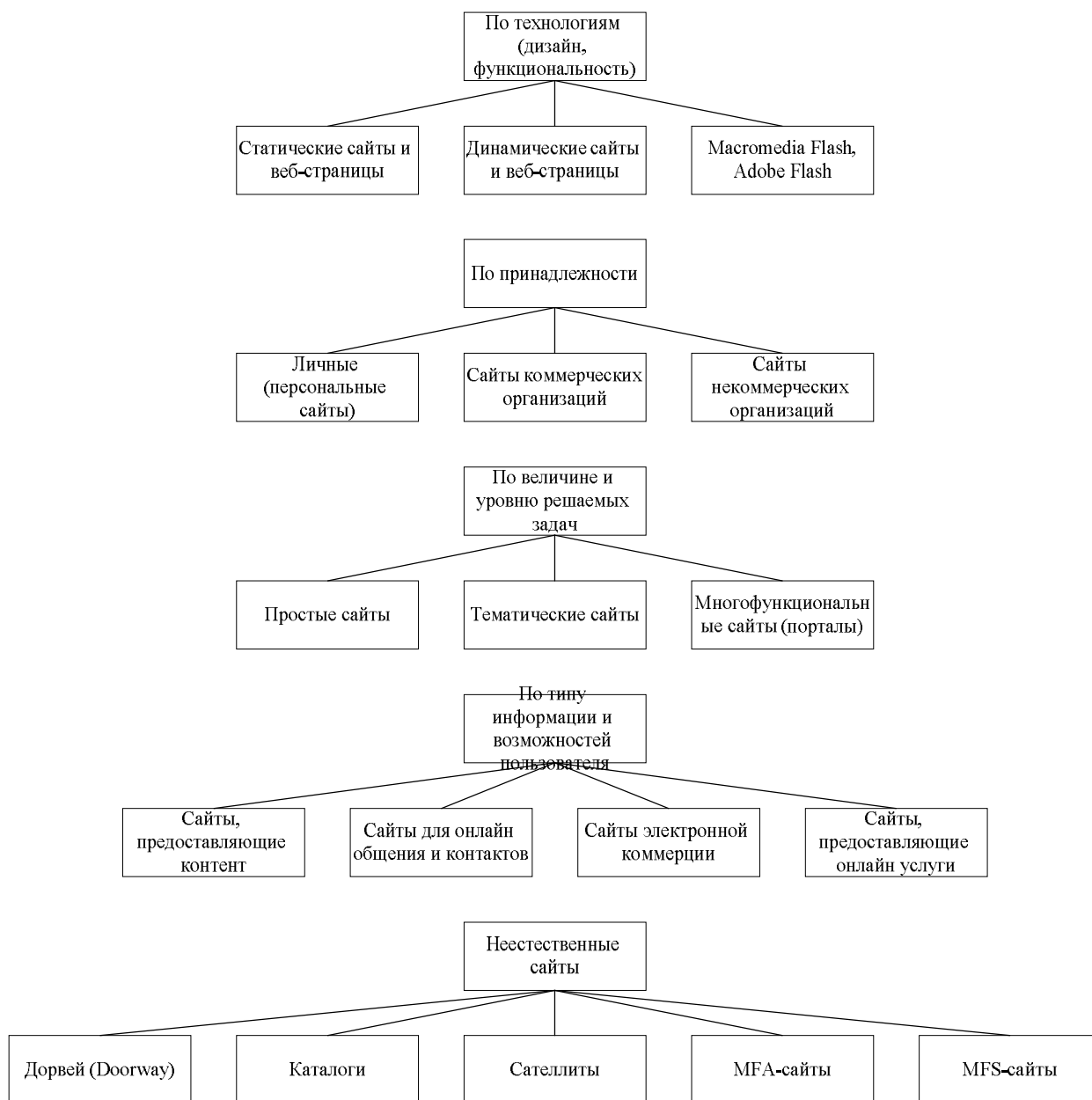


Рис. 1 – Общая классификация сайтов

## **Виды, типы, разновидности сайтов**

Сейчас в Интернете - глобальной, всемирной компьютерной сети существует огромное количество сайтов. Эти сайты самые разные и отличаются друг от друга по очень большому числу параметров.

Построить одну-единственную классификацию сайтов из-за того, что есть очень много признаков, по которым они отличаются не возможно. Опишем несколько классификаций сайтов, разделяя их на виды и типы каждый раз по разным признакам. Самое простое - разделить сайты по используемым в них технологиям, влияющим на возможности по созданию для них определённого дизайна и функциональности. Также по этому признаку можно классифицировать не только сайты, но и отдельные веб-страницы.

Первая классификация - **виды сайтов по технологиям, влияющим на дизайн и функциональность**. Вторая классификация сайтов - **классификация сайтов по их принадлежности**.

Третья классификация - **виды сайтов по величине, по уровню решаемых ими задач**.

Большинство сайтов предназначено для обычных пользователей сети, и спектр целей, с которыми тот или иной пользователь может прийти в интернет, очень широк. Это могут быть и поиск информации, причём по самым различным темам и самого различного свойства, и общение, и использование онлайн-сервисов и т.п. Для достижения этих целей пользователь должен найти подходящий сайт (или сайты), а владелец или владельцы сайта должны каким-то образом привлечь на свой сайт целевую аудиторию, т.е. тех, кому этот сайт был бы интересен.

Итак, четвёртая классификация – **виды сайтов по типу информации и возможностей, которые они предоставляют пользователям сети**.

Отдельно выделим и такие виды сайтов, которые изначально не предназначены для живых пользователей. Для чего же они предназначены? Об этом можно узнать, прочитав о классификации неестественных сайтов.

## **Виды, типы, разновидности сайтов и классификация сайтов по используемым технологиям**

Изначально единственной технологией, с помощью которой создавались сайты и веб-страницы, была технология html. Однако такие страницы и сайты выглядели бедно и не обладали интерактивностью, поэтому за короткое время, прошедшее от начала массового распространения интернета, появились и стали использоваться при создании сайтов многочисленные дополнительные технологии и языки.

Итак, классифицируем сайты по используемым технологиям:

**1) Статические сайты и веб-страницы.** Статические сайты и веб-страницы - это сайты, сделанные по классической технологии html. Веб-страницы таких сайтов написаны полностью на языке html и имеют расширение **.htm** или **.html**. Статическими они называются потому, что, набирая определённый адрес (URL), соответствующий сайту или его определённой странице, вы всегда будете видеть одно и то же содержание. Это потому, что статические html-страницы лежат на сервере в неизменном виде и по вашему запросу сервер просто берёт их и отправляет вам в браузер. С динамическими страницами, которые не существуют в заранее предопределённом виде на сервере, а формируются при помощи серверных скриптов, ситуация иная, и, набирая один и тот же адрес, вы, в зависимости от разных условий, можете получать совершенно разное содержание. Статические сайты имеют свои как плюсы, так и минусы. Плюсы статических сайтов - это простота их создания и нетребовательность к аппаратным ресурсам. Статические сайты не требуют поддержки скриптов и баз данных и могут быть размещены на абсолютно любом хостинге, при этом они не будут создавать почти никакой нагрузки на сервер, и с ними не возникнет проблема, когда сайт тормозит или выдаёт различные сбои при большом притоке посетителей. Статические сайты могут включать в себя графику, анимацию и javascript, они прекрасно подходят для предоставления пользователям информации, которая не должна, по крайней мере, часто и существенно изменяться. Основным минусом статических сайтов является отсутствие интерактивности. Кроме того, в таких сайтах сложнее изменить информацию, внешний вид и дизайн страниц.

**2) Динамические сайты и веб-страницы.** Как уже было упомянуто, динамические страницы не существуют в неизменном виде на сервере, а формируются при помощи скриптов. Содержание и вид динамической страницы с одним и тем же адресом может быть совершенно различным в зависимости от разных условий - например, в зависимости от времени, от конкретного пользователя, от введенного пользователем запроса и т. п. Скрипты, которые используются для формирования динамических веб-страниц, могут быть написаны на разных языках. Распространёнными языками веб-программирования являются, например, php, perl, asp и т.д. Такие сайты могут предоставлять пользователям интерактивность - например, вводить разные запросы и осуществлять поиск по сайту, отправлять на сайт и сохранять свою собственную информацию и осуществлять общение с другими пользователями (как, например, в гостевых книгах, форумах и т.п.) и многие другие функциональные возможности. Кроме того, ведение и обновление этих сайтов гораздо проще. Минусы динамических сайтов в том, что они требуют поддержку дополнительных технологий, создают повышенную нагрузку на сервер, их сложнее оптимизировать, а ещё использование на сайте скриптов означает потенциальную угрозу для безопасности.

**3) Флэш-сайты.** Технология флэш (flash) позволяет создавать очень красивые, интерактивные, со звуком и анимацией, сайты, которые выглядят на порядок эффектнее, чем обычные html-сайты. Однако сложность и трудоёмкость их изготовления, а также то, что страницы, созданные на флэш, имеют, как правило, большой вес и долго загружаются, что отпугивает часть пользователей, приводит к тому, что флэш-сайты не пользуются особой популярностью.

## **Виды, типы, разновидности сайтов и классификация сайтов по принадлежности**

Сейчас в интернете создать свой сайт может практически каждый. Только доменов второго уровня там уже многие миллионы, не говоря даже о сайтах, имеющих домены третьего уровня, которые во множестве расположены на бесплатных хостингах, таких, как narod.ru.

По принадлежности сайты подразделяются на:

**1) Личные (персональные) сайты.** Таким сайтом владеет и осуществляет его поддержку один человек. Такие сайты сейчас составляют большинство. Спектр этих сайтов весьма широк - от маленьких домашних страничек (home page), где есть немного информации о владельце и его увлечениях до очень известных и популярных ресурсов.

**2) Сайты коммерческих организаций.** Этим сайтам тоже сейчас очень много. По мере того, как интернет набирает популярность (а в западных странах к нему приобщены уже более половины населения), всё больше фирм и компаний рассматривают интернет как удобную среду если не для прямого ведения бизнеса и продажи товаров в онлайн-магазинах, то, по крайней мере, для рекламы своих товаров и услуг. По степени развитости присутствия коммерческой организации в интернете среди коммерческих сайтов можно выделить несколько подвидов:

**сайты-визитки**, содержащие лишь небольшую общую информацию о фирме, такие сайты обычно не обновляются;

**промо-сайты**, которые предназначены для презентации и продвижения различных товаров и услуг, на таких сайтах обычно оставляют контактные данные - адреса, телефоны, схемы проезда, предназначенные для потенциальных покупателей;

полноценные **сайты электронной коммерции**, на которых можно заказать услуги или купить товар, не вставая с кресла перед монитором.

**3) Сайты некоммерческих организаций.** Разнообразные некоммерческие организации также стремятся заявить о своём существовании в сети интернет. Многообразие подобных сайтов огромно. Тут и сайты какого-нибудь регионального отделения пенсионного фонда, и сайты политических партий и общественных движений, и сайты научно-исследовательских институтов и т.д. и т.п. Весьма популярны в интернете новостные ресурсы, ресурсы, представляющие онлайн-версии различных СМИ. Для двух особых категорий сайтов некоммерческих организаций, а именно сайтов государственных и образовательных учреждений в интернете даже выделены специальные доменные зоны: .gov и .edu. Причём, в отличие от популярных доменных зон .com, .org, национальных доменных зон и т.п., попасть туда может далеко не каждый желающий. Таким образом, расположение сайта в зоне .edu, например, - это гарантия солидности образовательного учреждения.

Виды, типы, разновидности сайтов и классификация сайтов по величине, уровню решаемых задач.

### **Виды, типы, разновидности сайтов и классификация сайтов по величине, уровню решаемых задач**

По величине и уровню решаемых задач сайты могут отличаться очень сильно. Есть сайты, состоящие из одной-единственной странички, а есть сайты, содержащие так много информации и требующие так много ресурсов, что для их поддержания используется с десяток отдельных высокопроизводительных серверов.

Приведём условную классификацию сайтов по этим параметрам.

**1) Простые сайты, содержащие немного информации и состоящие из нескольких страничек** ("сайты-визитки", домашние странички и т. п.). Основная цель их создателей и владельцев, как правило, - просто обозначить своё присутствие в интернете. Такие сайты обычно содержат лишь немного информации о владельце и совсем мало или вовсе не содержат какой-либо другой интересной для пользователей сети информации.

**2) Тематические, узконаправленные сайты.** Такие сайты могут быть довольно большими и подробно освещать какой-либо вопрос, предоставляя пользователям интернета, интересующимся определённой темой, возможность получить по ней хорошую и детальную информацию. Тематический сайт может рассказывать, например, об истории Древнего Рима, об отдыхе на Канарских островах, о выращивании и разведении кроликов и т.п. Многие из тематических сайтов занимают достойное место в интернете. Однако тематические, узконаправленные сайты могут привлекать и быть интересны лишь определённой категории пользователей.

**3) Многофункциональные сайты (порталы).** Особое место в ряду интернет-сайтов занимают монстры интернета - многофункциональные и многотематические сайты-порталы. Создавать, поддерживать и обеспечивать работу таких сайтов сложно, но зато такие сайты могут привлекать практически все категории пользователей и каждому быть интересны. Сайты-порталы содержат в себе массу самой разнообразной информации и функций, на них можно узнать и свежие новости, и прогноз погоды, и прочитать гороскоп, и завести почтовый ящик, и вести блог, и создать сообщество по интересам и т. д. и т. п. Такие крупные порталы, как, например, [Яндекс](#), [Рамблер](#) или [Мэйл.ру](#) известны практически каждому пользователю Рунета.

## **Виды, типы, разновидности сайтов и классификация сайтов по типу информации и возможностям, предоставляемым пользователям интернета**

Более-менее детальная классификация сайтов по типу информации и возможностям, предоставляемым пользователям интернета, очень обширна. Построить такую классификацию по этим признакам, которая точно распределила все сайты по подходящим для них категориям, довольно проблематично. Крупные каталоги интернет-ресурсов, такие как Яндекс-каталог, DMOZ, list.mail.ru и т.д., пытаются решать эту задачу по-разному. Множество аспектов и соображений, исходя из которых, можно выделять категории и подкатегории, оставляют много места для произвола.

Тем не менее, постараемся и здесь выделить типы и дать некую классификацию. Вначале выделим наиболее общие группы сайтов **по назначению**, соотнеся их с максимально общими целями, ради которых пользователи приходят в интернет. Изначально интернет был средой для получения информации. И сейчас сайты, направленные на предоставление разнообразной информации пользователям, являются наиболее многочисленной группой.

Первая группа - **сайты, существующие для предоставления пользователям информации, контента**. Помимо собственно поиска информации, люди часто приходят в интернет ради общения, ради поиска людей, близких по интересам и т.п.

Вторая группа - **сайты, направленные на организацию общения и взаимодействия между пользователями**. Вскоре после появления интернета в него пришёл бизнес, и появилась третья группа сайтов. Это - сайты, предназначенные для предоставления услуг, продажи товаров через интернет, **сайты, основной функцией которых является организация электронной коммерции**.

И, наконец, ещё одной важной группой сайтов, являются **сайты, предоставляющие различные онлайн-сервисы**.

Напишем теперь об этих группах сайтов поподробнее.

**1) Сайты, предоставляющие контент.** В этой группе, в свою очередь, можно выделить множество разных типов сайтов по разным признакам. Во-первых, **сам вид контента может быть различным** - т.е. это может быть как текст, так и фотографии, рисунки, всякие звуковые файлы, например, музыка популярных исполнителей, и видео. По **характеру предоставляемого контента** можно выделить сайты информационно-тематические, новостные, развлекательные, сайты-библиотеки, сайты-базы определённого рода документов, например, база рефератов, разнообразные сайты-справочники, онлайн-энциклопедии и словари, сайты-каталоги, обобщающие информацию о других сайтах и т.д. Ну и, конечно же, **по тематике**. Однако список возможных тематик был бы настолько большим, что мы не будем приводить его на этом сайте. Хорошая классификация различных сайтов по тематикам представлена в [Яндекс-каталоге](#).

**2) Сайты для онлайн-контактов и общения.** Список типов сайтов в этой группе также очень велик. Сюда можно отнести и "классические" формы организации общения, такие как форумы, чаты, доски объявлений, так и новые, которые можно условно отнести к группе т.н. "веба 2.0". Среди этих новых - популярные в последнее время "социальные сети", всякие блоги-сообщества в одном флаконе, "вопрос-ответные" проекты и т.п. Сюда нужно отнести и такие сайты, как службы знакомств, сайты, организующие общение

между людьми, ищущими в сети работу и работодателями, например, всякие биржи фрилансеров, сайты, на которых можно принять участие в онлайн-играх, некоторые сайты, предназначенные для поддержки общения пользователей между собой, с равной долей отнести и к онлайн-сервисам, например, это относится к сайтам icq или skype. Потенциал развития этого сектора очень велик, и не исключено, что новые интересные возможности для организации контактов и общения через интернет будут придуманы в самое ближайшее время.

### **3) Сайты электронной коммерции.** Какие типы сайтов выделяются в этой группе?

Конечно, самым многочисленным видом будут всякие интернет-магазины, через которые сейчас можно продавать практически всё, что угодно. Также велик список сайтов, предоставляющих различные платные услуги, сейчас с помощью интернета можно оплатить и мобильную связь, и консультацию психолога, и хостинг для сайта. В эту группу войдут и сайты электронных платёжных систем, сайты банков, предоставляющих возможность управления счетами через интернет, сайты обменных пунктов различных валют, сайты, дающие возможность играть на Форексе и т.п. В особую категорию здесь можно вынести сайты, построенные на основе т. н. партнёрских программ, т.е., проще говоря, клоны известных интернет-магазинов, продающие в интернете те же товары той же торговой структуры, и получающие на этом без особых усилий какой-то процент с продаж.

**4) Сайты, предоставляющие онлайн-сервисы.** Тут мы обнаружим также большое разнообразие. Некоторые сервисы можно отнести и к другим группам, например, сервисы, предоставляющие хостинг (как хостинг для сайтов, так и хостинг для гостевых, форумов, чатов, блогов, файлов и т.п.) Весьма распространёнными ещё со времён зарождения интернета являются сервисы бесплатной электронной почты. Вообще, многие онлайн-сервисы трудно классифицировать потому, что при всей их огромной популярности и значимости для интернета, главные сайты, предоставляющие их, существуют всего в нескольких экземплярах. Например, одними из важнейших сервисов, необходимость в которых возникла ещё на заре развития интернета, являются поисковые сервисы. Но при этом достаточно популярных среди них (в рунете) всего три. Самым первым поисковиком Рунета был Рамблер. Сегодня лучшим и крупнейшим из поисковых сервисов в Рунете является [Яндекс](https://yandex.ru), а в мировом интернете лидерство держит Гугл (Google). Другие интересные сервисы - это, например, сервисы, предоставляющие онлайн-перевод веб-страниц (скажем, [translate.ru](https://translate.ru)), сервисы, помогающие вам проверить текст на орфографические ошибки (скажем, [orfo.ru](https://orfo.ru)), сервисы, позволяющие собирать закладки на интересные вам сайты (скажем, [metogi.ru](https://metogi.ru) или Яндекс.Закладки) и т. д. и т. п. В последнее время, особенно в западном интернете появилась тенденция пытаться создавать онлайн-версии популярных оффлайновых программ, таких, как Word, Excel и т.п. Насколько успешной будет эта попытка переноса в онлайн работы с приложениями, покажет время.

**Неестественные сайты.** Что же это за феномен? Изначально все сайты, появляющиеся в интернете, были предназначены для живых пользователей сети. Какой был смысл тратить кому-то время, усилия и деньги на создание и размещение в сети сайта, который был бы никому не интересен?

Когда в сети обострилась конкуренция между сайтами и их владельцы стали бороться за вывод своего сайта в "топ", т.е. за то, чтобы пользователи находили их сайт на первых местах в выдаче поисковых систем, и особенно, когда в интернет пришли деньги, появились всякие нехорошие люди, которые стали замусоривать интернет никчёмными и никому (кроме них самих) не нужными сайтами.

Какие цели преследуют создатели этих никчёмных сайтов? Основных целей две. Во-

первых, это продвижение другого сайта, либо своего, либо того, владелец которого заплатил за продвижение. Во-вторых, это заработок, получаемый непосредственно с этих самых никчёмных сайтов.

## **Какие бывают неестественные и никчёмные сайты?**

Начнём с сайтов, создаваемых для раскрутки других сайтов.

1) **Дорвеи (вход).** Трудно раскрутить один сайт так, чтобы он занимал высокие позиции по множеству поисковых запросов. Поэтому, когда конкуренция в интернете обострилась, недобросовестные вебмастера, чтобы затащить как можно больше посетителей на свой сайт, стали создавать в большом количестве дополнительные сайты, оптимизированные под несколько ключевых слов или фраз, чтобы занять по этим фразам место в топе поисковиков. После того, как пользователь заходил на сайт-дорвей, то он не только не находил там, что искал, а зачастую сразу же автоматически перенаправлялся на главный сайт, тот самый, на который владелец хотел затащить посетителей. Поисковые системы объявили дорвеи, как средство обмана посетителей, вне закона и начали с ними беспощадную борьбу, в результате чего к сегодняшнему моменту пик дорвестроения спал, но они всё ещё иногда встречаются. Существуют даже специальные программы, которые позволяют генерировать дорвеи тысячами, заполняя их разнообразным (но оптимизированным под определённые запросы), бредом.

2) **Каталоги.** Изначально (когда сайтов в интернете было ещё немного) каталоги создавались с благой целью - дать ссылки посетителям каталога на интересные ресурсы, помочь им быстрее найти нужную информацию в интернете, нужный сайт. Полезные каталоги существуют и сейчас. Однако большая часть каталогов создаётся и используется совсем для других целей. Вебмастерам регистрация сайта в каталогах нужна, чтобы получить больше ссылок на свой сайт и, за счёт этого, приподнять его ближе к топу поисковиков. Владельцы же каталогов, в который они собирают все подряд сайты, зарабатывают на рекламе, либо создают много каталогов, а затем регистрируют в них за деньги чужие сайты. Раньше раскрутка сайта при помощи каталогов давала хороший эффект, но в последнее время этот эффект стал очень слаб. Определённой реинкарнацией каталогов стало появление нового типа каталогов - каталогов статей, которые заполняются уже не просто голыми ссылками, а многочисленными статьями, с проставленными внутри них ссылками на другие сайты.

3) **Сателлиты.** Сателлит - английское слово, которое в переводе на русский означает "спутник". Сателлит - это вспомогательный сайт, который не является самостоятельным ресурсом, а предназначен для продвижения главного сайта. Его отличие от дорвея в том, что главной его задачей является не сбор и перенаправление посетителей, а увеличение веса главного сайта при помощи проставленных на него ссылок. Сателлиты могут использоваться как для продвижения собственных сайтов, так и для продвижения чужих сайтов, когда ссылки с них продаются за деньги (в этом случае их можно отнести к категории MFS-сайтов).

Продолжим с сайтов, создаваемых теми, кто решил непосредственно с их помощью заполучить заработок в интернете.

4) **MFA-сайты.** Что такое MFA-сайты? MFA расшифровывается как "Made For AdSense", т. е. "Сделанные Для Адсенсе". AdSense - это популярная система контекстной рекламы

Google AdSense. Таким образом, MFA-сайты - это сайты, сделанные для заработка на контекстной рекламе (в рунете, кроме AdSense, имеющей заморочки с выводом средств, распространены и другие системы контекстной рекламы - Бегун и Яндекс.Директ). Конечно, нет ничего плохого, если владелец интересного, оригинального сайта получает дополнительный доход от этого сайта с помощью контекстной рекламы. Однако те, кто делает MFA-сайты, не заморачиваются с созданием нормальных сайтов, содержащих уникальную информацию и действительно нужных пользователям. Часто содержание таких сайтов представляет собой перемешанный копипаст с других сайтов или даже заимствуется с других сайтов и лент новостей в автоматическом режиме. В результате интернет заполняется, по сути, информационным мусором.

5) **MFS-сайты.** Что такое MFS-сайты? MFS расшифровывается как "Made For Sape", а Sape - это наиболее популярная в рунете биржа ссылок. Т. е. MFS-сайт - это сайт, предназначенный для продажи ссылок. В отличие от MFA-сайтов, для MFS-сайтов не нужно привлекать и заводить на сайт посетителей, которые кликали бы на контекстную рекламу, достаточно нарастить сайту "пузомерки", т.е. показатели ТИЦ и PR, чтобы была возможность продавать ссылки с него подороже.

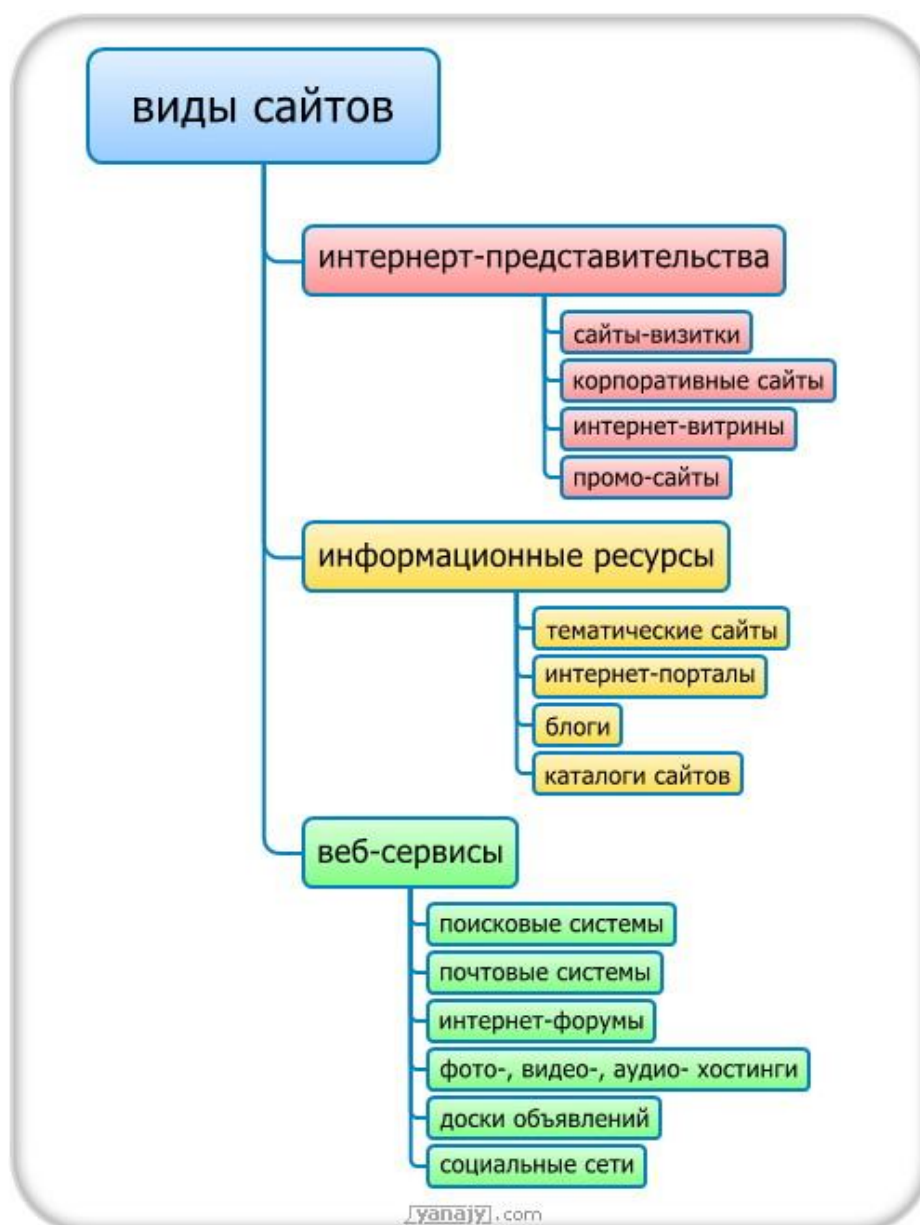


Рис. 2 – Узкая классификация сайтов

# **Контроль знаний**

## **1. Что такое сайт?**

- Совокупность электронных документов частного лица или организации в компьютерной сети, объединённых под одним адресом.
- Способ передачи информации разных типов
- Информационное пространство (узел) всемирной паутины (Интернета), содержащий контент. Контент – расширенная медиа-группа (медиа-группа + текст). Медиа-группа – аудио-, видео-файлы и изображения.

## **2. Общая классификация сайтов**

По технологии, по принадлежности, по величине и уровню решаемых задач, по информационной наполненности и возможностям пользователя, неестественные сайты.

## **3. Интернет- (вэб-) сервисы согласно узкой классификации сайтов**

Поисковые системы, почтовые системы, форумы, хостинги медиа-группы, доски объявлений, социальные сети.

## **4. Доменная структура адресов**

Домен – поименованное обозначение узла всемирной паутины (Интернета), хранящее информацию о его IP-адресе (Internet Protocol Address).

Доменное имя читается справа налево. Справа – верхний (I уровень), слева – нижний (N-уровень).

## **5. Классифицировать три наиболее посещаемых Вами лично сайта**

| №  | Название                                                        | По технологии        | По принадлежности        | По задачам   | По возможностям пользователя |
|----|-----------------------------------------------------------------|----------------------|--------------------------|--------------|------------------------------|
| 1. | <a href="http://vk.com">http://vk.com</a>                       | Динамический + Flash | Коммерческой организации | Портал       | С контентом + онлайн общение |
| 2. | <a href="http://google.com">http://google.com</a>               | Динамический         | Коммерческой организации | Портал       | С предоставлением услуг      |
| 3. | <a href="http://kinozal.tv">http://kinozal.tv</a>               | Динамический         | Коммерческой организации | Тематический | С контентом                  |
| 4. | <a href="http://twitter.com">http://twitter.com</a>             | Динамический         | Коммерческой организации | Тематический | Онлайн общение               |
| 5. | <a href="http://instagram.com">http://instagram.com</a>         | Мобильный сервис     | Коммерческой организации | Тематический | С контентом + онлайн общение |
| 6. | <a href="http://yandex.ru">http://yandex.ru</a>                 | Динамический         | Коммерческой организации | Портал       | С предоставлением услуг      |
| 7. | <a href="http://counter-strike.ru">http://counter-strike.ru</a> |                      |                          | Тематический | С контентом + онлайн общение |
| 8. | <a href="http://netpulse.ru">http://netpulse.ru</a>             | Статический сайт     | Коммерческой организации | Тематический | С предоставлением услуг      |

## **Лекция 2. Поисковые системы. Эффективный поиск в Интернет**

**Поисковые системы** (синонимы: поисковик, поисковый сервер, поисковая машина) - инструмент для поиска информации в Интернете.

Универсальная поисковая система – это комплекс программ и мощных компьютеров, выполняющих следующие функции:

специальная программа («паук») постоянно просматривает web-страницы и по ключевым словам составляет базы данных индексов.

Web-сервер принимает от пользователя запрос на поиск информации и передает его специальной программе – «поисковой машине».

Поисковая машина просматривает базу данных, составляет список страниц, удовлетворяющих условиям запроса, и возвращает его web-серверу.

Web-сервер оформляет результаты запроса и передает их на компьютер пользователя.

### **Технология эффективного поиска в Сети**

При поиске информации выделяют следующие критерии:

- полнота охвата информационного поля, в котором поиск происходит;
- достоверность полученного результата;
- скорость получения результата.

*Полнота* зависит от качества запроса, составленного пользователем и возможностей конкретной поисковой системы.

*Достоверность* результата зависит от особенностей функционирования поисковой системы, скорости обновления информации в предметной области, в которой осуществляется поиск

*Скорость* получения результата зависит от характеристики каналов связи, особенностей организации работы поисковика, и, тем не менее, от «качества» построения запроса

Если на работу поисковых систем пользователь непосредственно влиять не может, то качество составления запроса – целиком в его компетенции.

### **Приемы простого поиска**

#### **Поиск группы слов**

Слова «открытое» или «образование» дадут при поиске поодиночке большое число разнообразных ссылок, относящихся к совершенно различным темам, причем вряд ли имеющим отношение к «открытому образованию». Поэтому рекомендуется добавлять одно или два ключевых слова, связанных с искомой темой. Например, «открытое образование» или «технологии открытого образования». Необходимо также сужать область вопроса. Если необходимо найти информацию о правовой системе Гарант, то запрос « правовая система Гарант » выдаст более подходящие документы, чем просто «правовая система». Количество слов в группе не ограничивается.

#### **Поиск словоформ**

В большинстве случаев поисковая система по умолчанию ищет все словоформы языка. Однако, можно указать поисковой системе не перебирать все словоформы слов из запроса при поиске. Во многих системах для этого используется восклицательный знак.

Например, запрос «!кошкин» с большой вероятностью найдет страницы, цитирующие строчку из стихотворения «Кошкин дом».

#### **Роль прописных букв**

В случае если пользователь ввел в качестве запроса ключевое слово с прописной буквы, поисковая машина не найдет страниц, где содержится это слово, начинающееся со

строчной буквы. Поэтому заглавные буквы в запросе рекомендуется использовать только в именах собственных. Например, «город Москва», «Марк Тулий Цицерон».

#### Значение подстановочных символов

Когда нет уверенности в том, что поисковая система правильно обрабатывает словоформы (то есть когда речь идет, например, об именах собственных или словах иностранного происхождения) поисковые системы позволяют использовать подстановочные символы. Чаще всего это символ «\*» вместо любого количества любых символов до конца слова. Например, если пользователь хочет найти страницы, содержащие слова «республика Татарстан», но устроит и Татарская республика, тогда надо подать запрос «республика Татарс\*».

#### Учет зарезервированных слов

Зарезервированными словами (стоп-словами) считаются те слова, которые не учитываются при поиске. Обычно к ним относятся все короткие слова, в которые входят менее 4 букв (предлоги, союзы и т.п.). Например при запросе «мы в Италии» будут найдены документы, в которые входит слово «Италии» или его словоформы.

#### Средства контекстного поиска

Если ключевые слова взять в кавычки, то поисковая система должна найти документы, в которых данная фраза присутствует буквально (поиск цитаты).

### Приемы расширенного поиска

Для более быстрого и успешного поиска в поисковых машинах совместно с ключевыми словами используются различные логические операторы. Благодаря этому можно сконструировать запрос так, что будут найдены не сайты на интересующую тему, а конкретные страницы и даже отдельные документы. Правила составления сложных запросов на одной поисковой машине могут отличаться от таковых на другой, но в любом случае будут использоваться следующие основные операторы:

#### Оператор И (AND)

С помощью этого оператора объединяют два или более слов так, чтобы они все присутствовали в искомом документе. Часто вместо И используют & или +. Пример: по запросу *юрист И программа* будут найдены документы, содержащие и то и другое слово.

#### Оператор ИЛИ (OR)

Обеспечивает поиск по любому из слов группы. Пример: по запросу *образование ИЛИ обучение* будут найдены документы, содержащие слово *образование* или *обучение*.

#### Логические скобки

Применяются, когда надо управлять порядком следования логических операторов.

Пример: по запросу *Ломоносов ИЛИ (Михаил И Васильевич)* будут найдены документы, содержащие слова *Ломоносов* или *Михаил И Васильевич*.

#### Оператор НЕ (NOT)

Используется, когда из результатов поиска надо исключить какое-либо ключевое слово. например по запросу *правоведы НЕ адвокаты* будет найдена информация о правоведах, не являющихся адвокатами.

#### Оператор БЛИЗКО (NEAR)

Поиск с указанием расстояния. Он позволяет указать, на каком расстоянии друг от друга могут располагаться слова в документе. Синтаксис такого запроса различен у разных поисковых систем.

### Средства специального поиска

Позволяют выполнять поиск документов по:

датам;

по ссылкам на определенный адрес;

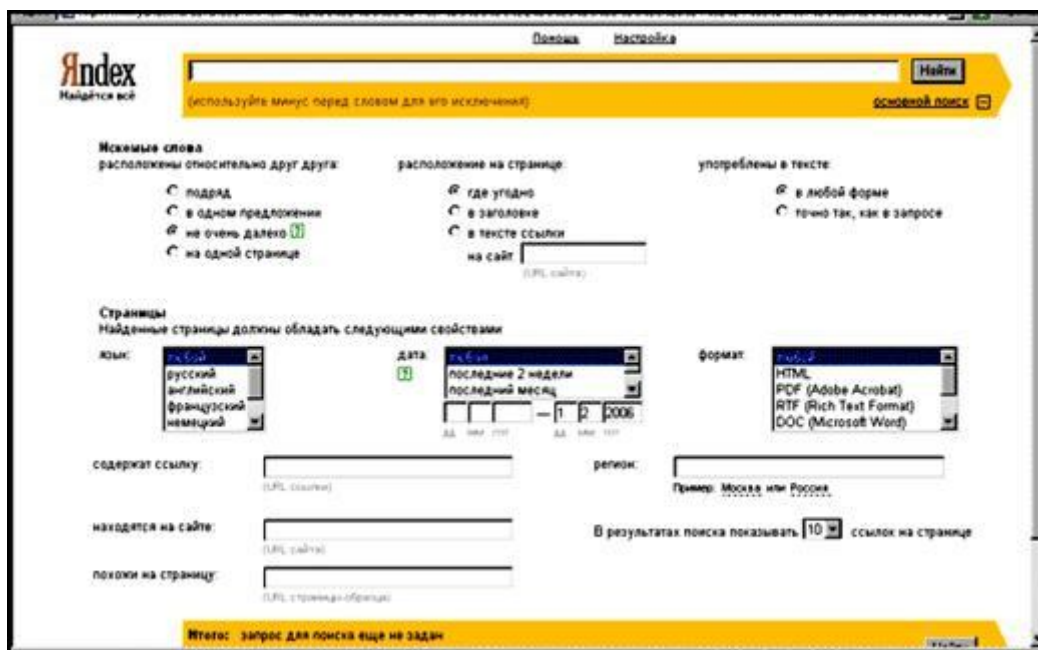
по заголовкам веб-страниц;  
по доменному имени сервера;  
и др.

Все средства поиска реализуются при помощи так называемого языка запросов. Языки запросов разных поисковых систем незначительно отличаются друг от друга. Для примера приведем синтаксис языка запросов поисковой системы Yandex.

#### Синтаксис языка запросов поисковой системы Yandex

| Синтаксис       | Что означает оператор                                                                        | Пример запроса                                                                         |
|-----------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| пробел<br>или & | логическое И (в пределах предложения)                                                        | лечебная физкультура                                                                   |
| &&              | логическое И (в пределах документа)                                                          | рецепты && (плавленый сыр)                                                             |
|                 | логическое ИЛИ                                                                               | фото   фотография   снимок   фотоизображение                                           |
| +               | обязательное наличие слова в найденном документе (работает также в применении к стоп-словам) | +быть или +не быть                                                                     |
| ( )             | группирование слов                                                                           | (технология   изготовление)<br>(сыра   творога)                                        |
| ~               | оператор И НЕ (в пределах предложения)                                                       | банки ~ закон                                                                          |
| ~~ или -        | оператор И НЕ (в пределах документа)                                                         | путеводитель по Парижу ~~<br>(агентство   тур)                                         |
| /(n m)          | расстояние в словах (-назад +вперед)                                                         | поставщики /2 кофе<br>музыкальное /(-2 4)<br>образование<br>вакансии ~ /+1 обучающихся |
| « «             | поиск фразы                                                                                  | «красная шапочка»<br>(эквивалентно красная /+1 шапочка)                                |
| &&/(n m)        | расстояние в предложениях (-назад +вперед)                                                   | банк && /1 налоги                                                                      |

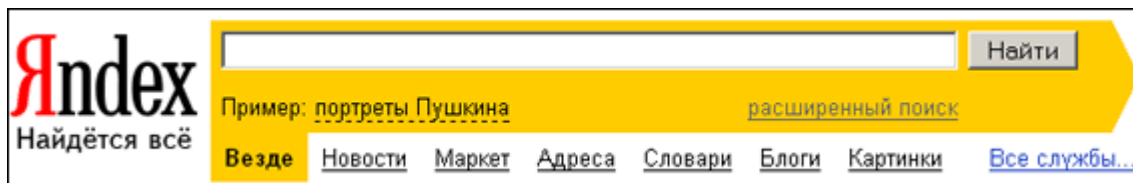
**Пользуйтесь расширенным поиском и языком запросов конкретной поисковой системы**



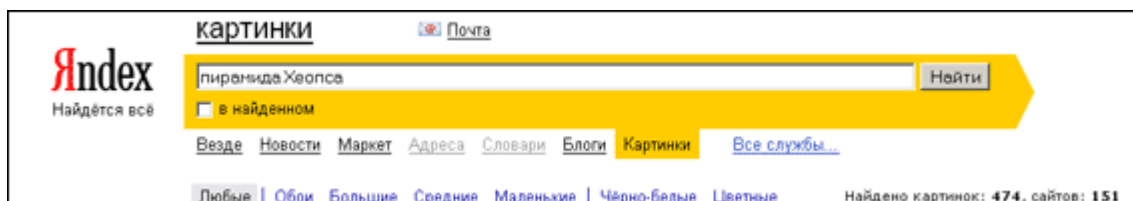
Рассмотрим наиболее популярные поисковые системы и те дополнительные услуги, которые они могут предложить:

#### 1) www.yandex.ru

Поисковая система Yandex предлагает широкий спектр поисковых возможностей: поиск по сайтам и лентам новостей, поиск товаров (с последующей возможностью отбора по каталогу), поиск учреждений по адресам (совместно с [YELLOWPAGES.RU](#)), поиск по энциклопедиям и словарям (в том числе – перевод слов), поиск по блогам и форумам, поиск изображений с указанными параметрами.

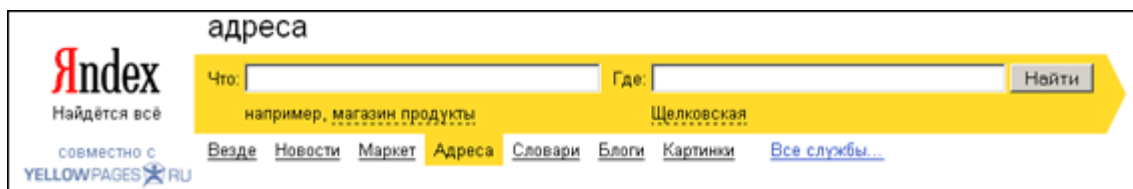


В некоторых случаях, при выборе одной из закладок, появляется возможность задать более точные параметры поиска.

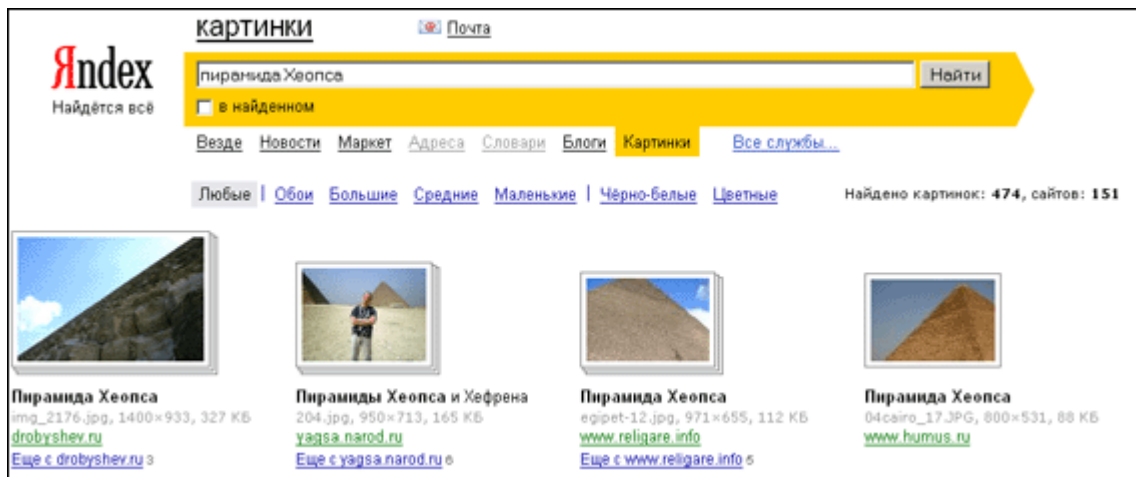


Например, при поиске картинки можно указать ее примерный размер и вид (цветная или черно-белая).

На закладке «Адрес» поисковая строка принимает иной вид (что логично):



Отображение ссылок на найденные объекты также будет различаться в зависимости от типа информации. Например, найденные картинки будут представлены в виде галереи (в формате «preview», последующее нажатие на картинку приводит к ее открытию в полный размер).



## 2) www.mail.ru

Поисковая система Mail.ru наследует возможности системы Google. Однако результат поиска представлен несколько иначе.

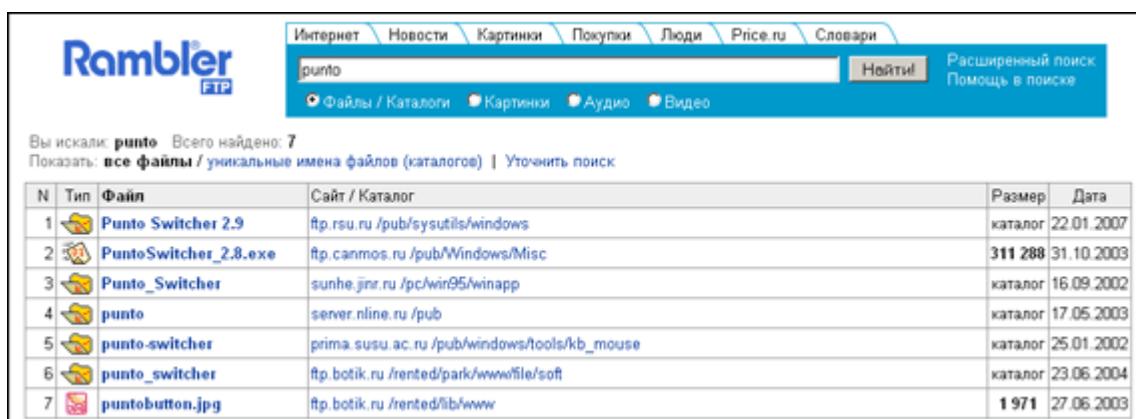


## 3) www.rambler.ru

До последнего момента поисковая система Rambler предлагала довольно ограниченный набор возможностей, даже поиск изображений был для нее не доступен. Однако в последние месяцы она претерпела значительные изменения, предложив наряду с традиционными для современных поисков услугами и что-то свое:

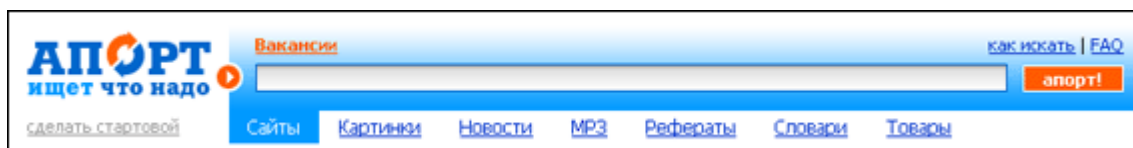
Появившаяся услуга поиска «Люди», в качестве источника использует базу данных национальной службы взаимного поиска людей «Жди меня». Возможен поиск в Большом справочнике по ценам Price.ru.

Однако некоторые полезные опции, например – поиск файлов, были вынесены в раздел «Все виды поиска».



Результаты поиска *файлов* по запросу «*punto*»:

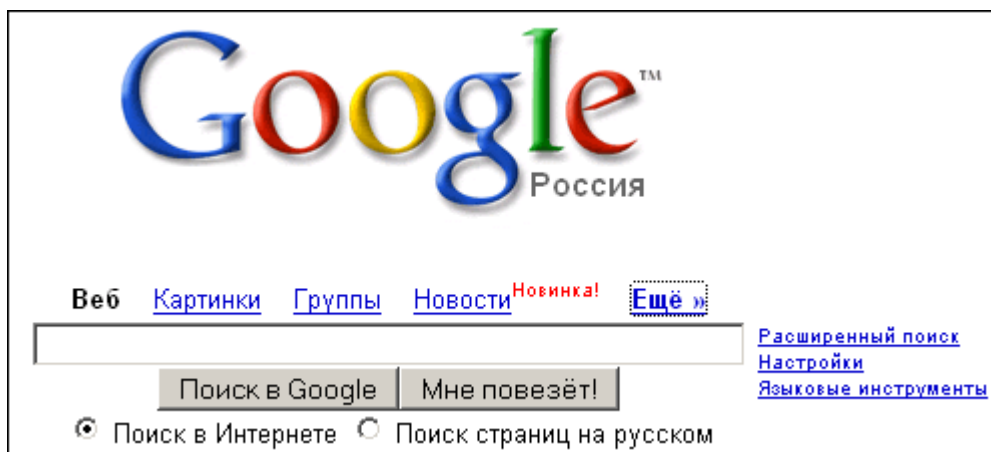
## 4) www.aport.ru

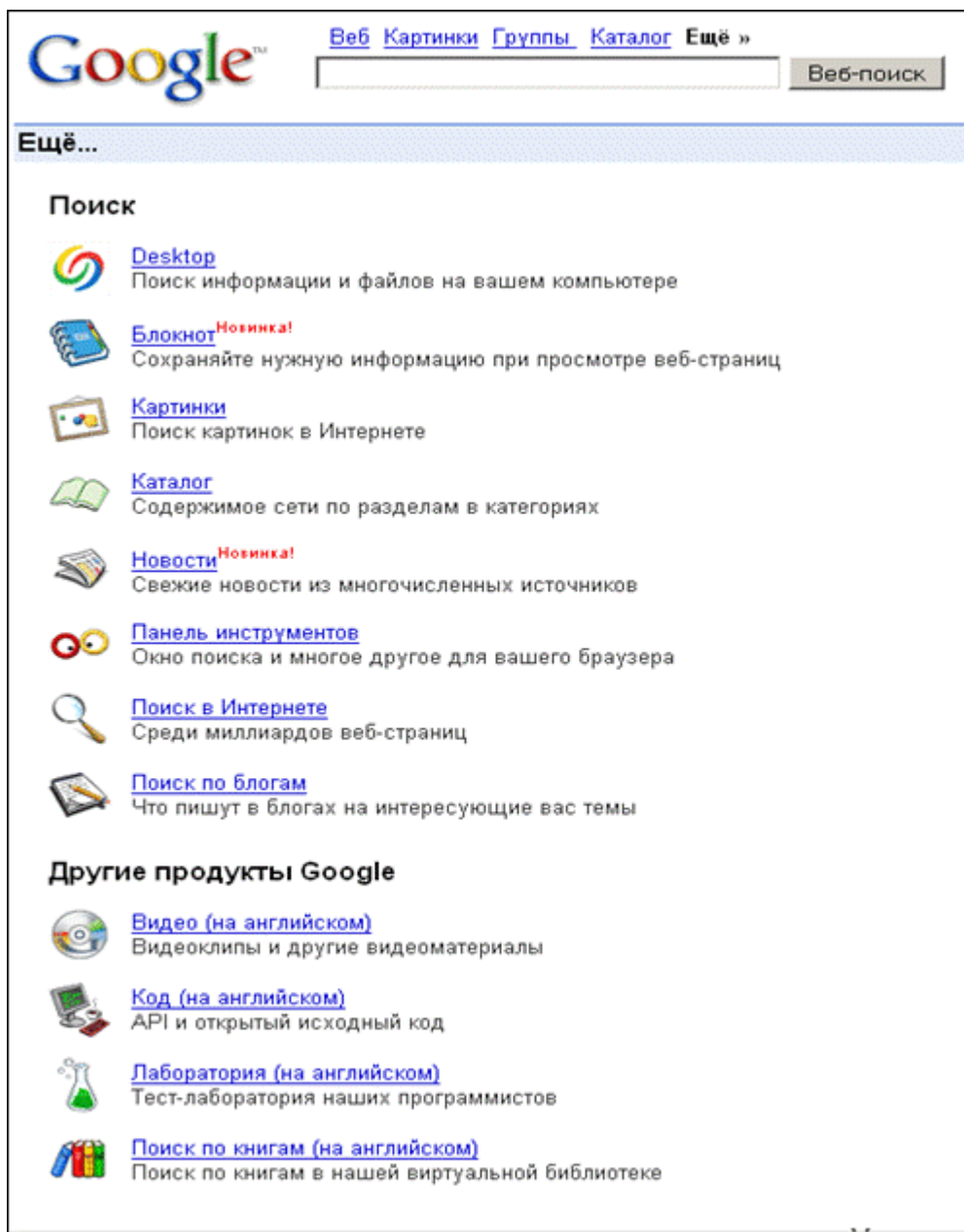


Апорт часто воспринимают как молодежный поисковик. Он обладает специфическими возможностями – отдельно предлагается поиск mp3-файлов, рефератов, вакансий.

#### 5) [www.google.ru](http://www.google.ru)

Русская версия мирового гиганта Google кроме стандартных вариантов предлагает выбрать дополнительные функции с помощью ссылки «Ещё».





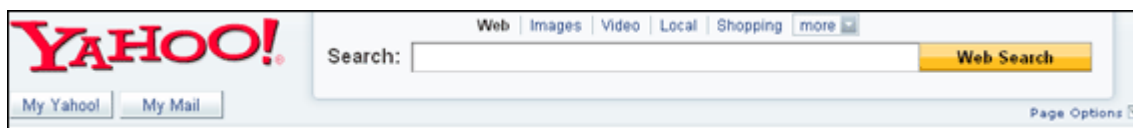
Анализируя предлагаемые возможности видно, что многие из них компания Google еще не адаптировала для русскоязычной части Интернета. «Динозавры» мировых поисковых систем Yahoo и AltaVista отличаются от русскоязычных возможностью поиска видео.

6) [www.altavista.com](http://www.altavista.com)

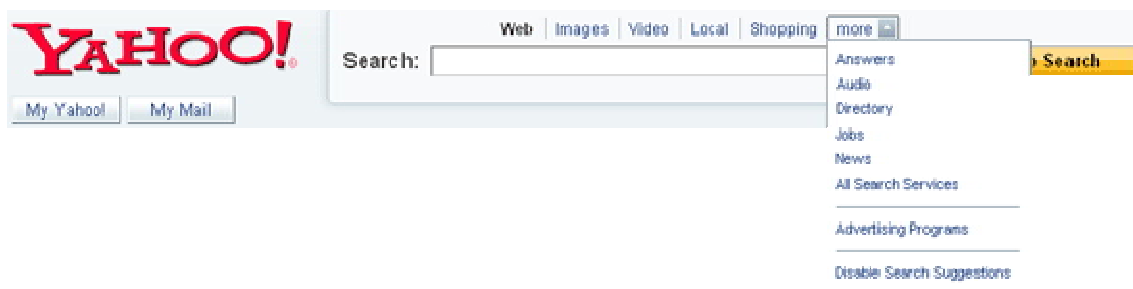


Интересна возможность выбрать материалы только на английском языке и очертить страну поиска.

Дополнительно можно указать тип разыскиваемого файла, регион поиска по домену, временной период.  
7) [www.yahoo.com](http://www.yahoo.com)



Дополнительные возможности на Yahoo выбираются с помощью выпадающего списка «More»:



Популярные поисковые системы:

<http://www.yahoo.com>

<http://www.altavista.com>

<http://www.google.com>

<http://www.yandex.ru>

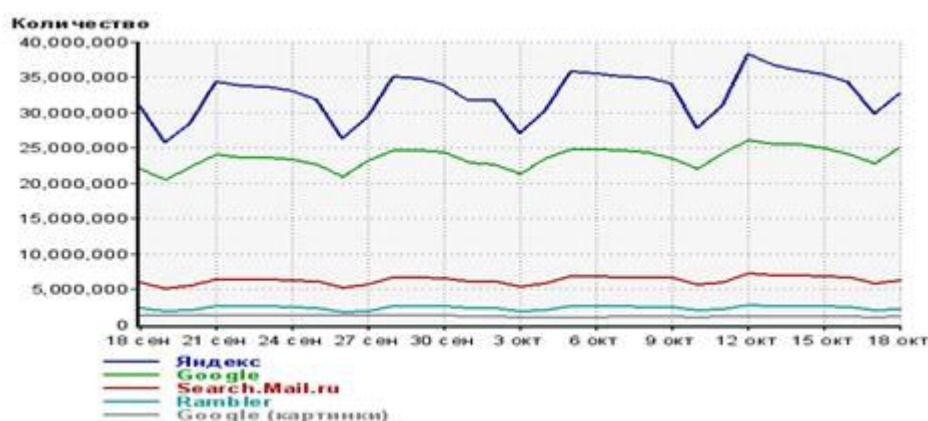
<http://www.aport.ru>

<http://www.rambler.ru>

<http://www.google.ru>

<http://www.liveinternet.ru/stat/ru/searches.html> - рейтинги русскоязычных поисковых систем (по дням, неделям, месяцам).

И в заключении, обратите внимание на рейтинги русскоязычных поисковых систем и сделайте вывод:



## **Лекция 3. Продвижение услуг в Интернете**

### **Этапы размещения рекламных сообщений**

Создание стандартного рекламного объявления в Google AdWords занимает, как правило, менее 30 минут. Процесс размещения рекламы происходит в 9 шагов.

**На первом** рекламодателю необходимо создать аккаунт в Google AdWords, для чего достаточно указать адрес электронной почты и пароль, используемые для доступа к другим сервисам Google. При отсутствии таковых, можно зарегистрировать аккаунт AdWords на любой другой адрес электронной почты.

**На втором этапе** рекламодателю предлагается выбрать целевую аудиторию (т.е. произвести [таргетинг](#)), путём выбора местоположения (страна, регион, город), языка потенциальных клиентов, а также демографических показателей (возраст и пол).

**На третьем этапе** необходимо указать бюджет рекламной кампании.

**На четвёртом** устанавливается цена за клик по рекламному сообщению. При выборе рекомендованного варианта «Автоматическое назначение ставок» указывается верхний предел цены за клик, к примеру, 10 руб., а система автоматически подбирает минимально необходимую цену за клик для показа рекламного объявления на лучшей в каждом конкретном случае позиции. При этом фактическая цена за переходы в большинстве случаев меньше максимальной цены и никогда её не превышает.

**На пятом** этапе рекламодатель может создать группу объявлений, позволяющую объединить несколько вариантов объявлений, посвященных одному товару или услуге. AdWords автоматически поочерёдно показывает разные объявления из группы, выявляя при этом наиболее [эффективные](#).

**Шестой этап** посвящён созданию непосредственно рекламного объявления, для чего необходимо ввести заголовок объявления (не более 25 символов), кратко описать продукт (2 строки не более чем из 35 символов каждая) и указать адрес веб-страницы, на которую перейдет пользователь, нажавший на контекстное объявление. Внести изменения в текст объявления возможно в любой момент.

**На седьмом этапе** рекламодателю предлагается выбрать ключевые слова, при использовании которых в поисковом запросе пользователя будет показано размещаемое объявление. Предлагаемый на этом этапе инструмент подсказки ключевых слов позволяет подобрать ключевые слова, проверить их популярность (количество запросов пользователей по ним) и узнать уровень конкуренции по данным словам среди рекламодателей.

**На восьмом этапе** можно создать дополнительные варианты объявлений, из которых впоследствии AdWords эмпирическим путём отберёт наиболее эффективные.

**На заключительном, девятом, этапе** рекламодателю необходимо выбрать способ оплаты и ввести платежные данные. Среди способов оплаты имеются: банковский перевод, кредитные карты [Visa](#) и [MasterCard](#), [Webmoney](#), [Яндекс.Деньги](#), причём [Webmoney](#) и [Яндекс.Деньги](#) доступны только для жителей России. По завершению проверки объявления модератором производится оплата в соответствии с требованиями Google AdWords, и интернет-реклама появляется в сети.

### **Редакционные правила**

Определив, на кого будет направлена реклама и выбрав соответствующие [ключевые слова](#), веб-издатель создаёт [текстовые объявления](#), которые информируют его потенциальных клиентов (целевую аудиторию) о предлагаемых им продуктах или услугах. Для создания рекламного сообщения существуют определённые *редакционные*

*правила* Google AdWords, которые касаются описания сайта в рекламном сообщении и уникальной выгоде предлагаемого продукта или услуги. Рекламные объявления, размещаемые в системе AdWords, кратки и идентичны по структуре: заглавие-ссылка, адрес сайта и текстовый блок размером не более трёх строк.

## Работа с ключевыми словами

Для выбора оптимального множества ключевых слов AdWords обладает [инструментом подсказки ключевых слов](#). Функции данного инструмента: сортировка списка ключевых слов (по распространённости, истории эффективности в AdWords, по цене и прогнозируемой позиции объявления на странице результатов поиска Google);

упрощенная работа с ключевыми словами (возможность сочетания различных источников, загрузки из [.csv](#)-файла);

три варианта поиска ключевых слов (по введённым ключевым словам, по словам с высоким рейтингом кликов или по URL любой веб-страницы);

предоставляемая статистика регулярно обновляется.

Создатель рекламного сообщения также имеет возможность изменять **параметры соответствия ключевых слов** для возможности его показа наиболее подходящим клиентам, а также для снижения цены за клик (CPC), либо цены за тысячу показов (CPM), а также увеличения рентабельности инвестиций (ROI). В данный момент существует четыре метода ориентации объявлений AdWords на пользователей.

**Широкое соответствие** (установлено по умолчанию). Пример: Если в список ключевых слов включить *теннисная обувь*, то объявления показываются в ответ на запрос по словам *теннисный* и *обувь*, причём слова могут быть указаны в любом порядке и, возможно, в сочетании с другими словами.

**Фразовое соответствие**. Работает, если заключить ключевую фразу в кавычки.

Пример: «*теннисная обувь*», объявление будет показано в ответ на поисковый запрос по словосочетанию *теннисная обувь* именно с таким порядком слов (*мягкая теннисная обувь*, но не *обувь для тенниса*). Это соответствие более точно ориентировано, чем Широкое, но всё же более свободно, чем Точное соответствие.

**Точное соответствие**. Если ключевая фраза заключена в квадратные скобки.

Пример: [*теннисная обувь*], объявление будет показано только в ответ на поиск по фразе *теннисная обувь* с точным соблюдением порядка слов, и без других слов в запросе (запрос *мягкая теннисная обувь* не сработает). Это самый целенаправленный вариант соответствий.

**Минус-слова**. Заносятся в отдельные наборы, на уровне Группы объявлений или Кампании (возможно вместо отдельных слов использовать их именованные списки). В отличие от Яндекс.Директ, нельзя минусовать сам запрос, поскольку эффект будет прямо противоположным. Пример: Если выбрать ключевую фразу *теннисная обувь -размеры*, то минус будет проигнорирован как спецсимвол, а объявление будет показано по запросу *теннисная обувь размеры*.

Важно: в минус-словах на Поиске отдельно учитываются все словоформы, т.е. помимо слова *размеры* нужно также минусовать *размер*, *размера*, *размеров* и другие его варианты (по крайней мере часто встречающиеся).

Есть возможность использования отдельных URL-адресов издателя для каждого ключевого слова, или релевантных ключевым словам разделов сайта.

## Система оплаты

До 2002 года вся контекстная реклама Google AdWords продавалась по системе CPM (cost-per-thousand impressions: «оплата за просмотры»), однако постепенно был совершён переход на модель CPC (cost-per-click: «оплата за переходы»).

## Отчёты

Статистика учётных записей Google AdWords отражается в отчётах на уровнях рекламной кампании, группы объявлений, ключевого слова или сайта. Она включает:

- количество кликов (clicks),
- показов (impressions),
- рейтинг кликов (CTR),
- среднюю цену за клик или за тысячу показов (average CPC or CPM),
- стоимость (cost),
- среднюю позицию (average position),
- показатель переходов (conversion rate), а также
- соотношение между стоимостью и переходом (cost-per-conversion).

Веб-издатель может сам создавать отчёты с интересующей его статистикой от уровня кампании до уровня сайта. Также можно воспользоваться сервисом [Google Analytics](#), который анализирует, какие страницы посещают читатели на сайте веб-издателя. Таким образом, Google AdWords предоставляет все необходимые инструменты и формирует данные, позволяющие контролировать и изменять ход рекламной кампании для достижения её максимальной эффективности.

Таргетинг (англ. target - цель) - рекламный механизм, позволяющий выделить из всей имеющейся аудитории только ту часть, которая удовлетворяет заданным критериям (целевую аудиторию), и показать рекламу именно ей.

## Виды таргетинга

Подбор рекламных площадок. Наиболее популярный вид таргетинга. Осуществляется путем подбора рекламных площадок так, чтобы их посетители соответствовали целевой аудитории.

Тематический таргетинг. Показ рекламы на веб-сайтах, соответствующих определенной тематике.

Таргетинг по интересам (контекстная реклама). Показ рекламы в соответствии с интересами посетителей рекламной площадки.

Географический таргетинг (геотаргетинг). Показ рекламы целевой аудитории, ограниченной некоторым географическим регионом, выбранным рекламодателем.

Таргетинг по времени показа (утро или вечер, будни или выходные). Позволяет ограничить показ рекламы по времени в течение дня, недели, года.

Социально-демографический таргетинг - по возрасту, полу, доходу, должности и т. д.

Ограничение количества показов одному пользователю, позволяет регулировать количество показов рекламного носителя одному уникальному пользователю в процессе его взаимодействия с рекламной площадкой. Чаще всего применяется в баннерной рекламе с оплатой за 1000 показов.

Поведенческий таргетинг. Самое перспективное направление на сегодняшний день. Суть его сводится к внедрению механизма сбора информации о действиях пользователя в интернете с помощью cookie-файлов. Информация собирается в так называемых профилях и содержит данные о просмотренных сайтах, поисковых запросах, покупках в интернет-магазинах и т. д. Получив такой профиль, рекламная служба может четко представить

себе портрет объекта, узнать его привычки и пристрастия, стать владельцем контактных данных.

Геоповеденческий таргетинг. Понятие впервые упомянуто компанией AlterGeo. Суть сводится к тому, что зная о перемещениях субъекта (точное местоположение вплоть до заведения, в котором находится субъект, умеют определять современные гео-социальные сервисы. Например, AlterGeo) и остановках ("чекинах" или "check-ins") в некоторых точках, можно четко представить привычки и пристрастия субъекта. Пример №1: если субъект ходит в пивные, то и пиво наверняка его интересует. Пример №2: Если субъект ходит в заведение определенного типа, то и другое заведение этого же типа наверняка его интересует.

Психологический таргетинг. Также одно из самых перспективных направлений. На основании информации о психологических качествах (психотип, социотип) выдается различная реклама. Теоретическое обоснование - Соционика

MSB (Matrix Shop Behaviours). Новейшая технология, суть которой заключается в следующем: Для каждого заказчика, на основе специфики общения его менеджеров с клиентами, создается поведенческая матрица, которая содержит в себе не только стереотипные модели мышления покупателей в данной бизнес-модели, но и весовые коэффициенты их значимости в процентах. Такая детализация позволяет оценивать качество сайта (структура, юзабилити, дизайн, содержимое текста), не на уровне хорошо/плохо, а на уровне конкретных цифр.

## **Прямой и косвенный таргетинг**

Прямой таргетинг нацелен на выбор целевой аудитории, напрямую интересующейся предлагаемым товаром или услугой. Косвенный таргетинг нацелен на аудиторию, являющуюся целевой для взаимосвязанных с предлагаемым видом товаров или услуг.

## **Анализ поисковых запросов**

Уникальную возможность таргетинга предоставляют поисковые системы, так как их посетители четко формулируют свою потребность в виде запроса, а также автоматически предоставляют системе некоторые свои характеристики.

**Google AdSense** — сервис [контекстной рекламы](#) от [Google](#). Программа автоматически размещает на [веб-сайтах](#) [текстовые](#) и [графические](#) объявления, подходящие по [контексту](#). Владельцы сайтов, размещающие рекламу, получают доход за клики (переходы посетителей по рекламным ссылкам) и показы рекламы (**Google AdSense для контекста**). Кроме того, издатели веб-сайтов с помощью AdSense могут предоставлять своим посетителям поиск Google как по Интернету, так и непосредственно по сайту, зарабатывая деньги от объявлений Google (**Google AdSense для поиска**).

Данный рекламный сервис особенно важен для обеспечения доходности небольших веб-сайтов, у которых отсутствуют средства для развития собственной системы продажи рекламы.

С помощью своей интерактивной учётной записи веб-издатель в любое время может просматривать количество [запросов](#), [кликов](#), рейтинг кликов, а также свои текущие доходы от использования AdSense.

В 1-ом квартале 2011 года на долю AdSense приходилось 28% доходов Google, что составляло 2,43 млрд долл.

## **Инструменты**

1. **Фильтр объявлений конкурентов.** Имеется возможность заблокировать объявления конкурентов.

2. **Просмотр объявления.** Изучение объявления (ручное и машинное) по некоторым характеристикам.
3. **Фильтр на контексты.** Отфильтровка объявлений, неподходящих для некоторых страниц.
4. **Выбор собственных объявлений.** В случае, если Google автоматически не может подобрать контекстное объявление, у веб-издателя есть возможность собственного выбора рекламных объявлений Google.

Google AdSense перечисляет деньги участникам программы раз в месяц, при накоплении суммы \$100. В зависимости от страны доступны разные методы оплаты: [чеком](#), через систему электронных платежей (EFT) непосредственно на банковский счёт, [Western Union](#) Quick Cash и через систему [Рапида](#). Обналичивание [чеков](#) для граждан, проживающих на территории стран [СНГ](#), возможно через банковские учреждения и некоторые [платёжные системы](#). Однако нужно иметь в виду, что в соответствии с Условиями предоставления услуг компания Google оставляет за собой право действовать по собственному усмотрению при выявлении недействительных кликов. На практике при выявлении так называемой "недопустимой активности" аккаунт Google AdSense блокируется и деньги за клики по рекламе на своём сайте Вы, скорее всего, не получите.

#### Типы объявлений

- Текстовые и графические объявления
  - Доска почёта 728×90
  - Баннер 468×60
  - Средний прямоугольник 300×250
  - Широкий небоскрёб 160×600
  - Небоскрёб 120×600
- Только текстовые объявления
  - Большой прямоугольник 336×280
  - Квадрат 250×250
  - Полубаннер 234×60
  - Малый прямоугольник 180×150
  - Кнопка 125×125
  - Вертикальный баннер 120×240

#### Ограничения на размещение рекламы

Владельцы веб-сайтов могут размещать не более трёх блоков рекламы на страницу. Стоит отметить, что не все сайты допускаются к участию в программе AdSense: в регистрации отказывают сайтам с порнографическим содержанием, сайтам, распространяющим идеи экстремизма и терроризма, а также содержащим информацию о наркотических веществах и алкогольных напитках. В некоторых странах, например в Китае и Индии, требуется, чтобы издатели владели своими сайтами не менее 6 месяцев. Это необходимо для обеспечения высокого качества рекламной сети Google и защиты интересов рекламодателей и издателей.

## **Лекция 4. Рекламные технологии на примере**

### **Яндекс.Директ**

#### **Что такое Яндекс.Директ**

**Яндекс.Директ** - это инструмент для размещения **контекстных** рекламных объявлений на страницах Яндекса и на сайтах — участниках его Рекламной сети. Реклама в Яндекс.Директе размещается с оплатой за клик по объявлению, а не за его показ. Реклама в Яндекс.Директе эффективна оттого, что ее видят именно те, кто ищет в интернете информацию о товарах или услугах — то есть, потенциальные клиенты. Где показываются объявления

**Страница результатов поиска Яндекса** — объявления показываются либо над результатами поиска, либо справа от них в ответ на поисковый запрос пользователя.

**Страница "Все объявления"** — объявления показываются на странице со всеми объявлениями Яндекс.Директа по конкретному запросу.

**Рекламная сеть Яндекса** — объявления показываются на страницах качественных и посещаемых сайтов Рунета.

**Площадки Яндекса** — объявления показываются на страницах Каталога Яндекса и на страницах результатов поиска по Каталогу, на страницах результатов поиска по Каталогу организаций и на страницах результатов поиска по блогам.

Подробнее об условиях показа объявлений смотрите правила показа.

## Сколько стоит реклама в Яндекс.Директе

Минимальная сумма заказа в Яндекс.Директе составляет 300 руб. Показы объявлений начинаются после оплаты и зачисления денежных средств на заказ.

Оплаченные средства будут расходоваться по мере того, как пользователи будут переходить на рекламируемый сайт/виртуальную визитку. То есть, рекламодатель оплачивает только фактические клики (переходы) по его объявлению.

Цена клика индивидуальна (от 30 коп.) для каждой ключевой фразы и может изменяться в зависимости от спроса на выбранные фразы со стороны других рекламодателей, размещающих объявления в том же регионе.

Исходя из параметров конкретного заказа (ключевых фраз, региона и стратегии показа) система прогнозирует\* сумму бюджета, которой должно хватить ориентировочно на неделю работы заказа. В момент выставления счета система показывает эту сумму. В любом случае вы можете оплатить рекомендуемую сумму или указать свою (от 300 рублей).

На Директе предусмотрена система скидок. Скидки рассчитываются автоматически от совокупного объема услуг, оказанных и оплаченных клиентом в течение последних 12 месяцев до момента выставления счета. Скидки являются накопительными, то есть, при достижении определенной границы отпускная цена уменьшается. Подробнее о размере скидок написано на странице о скидочной политике Яндекса.

Обратите внимание, что агентства скидок на контекстную рекламу в Яндекс.Директе не дают, поэтому любые предложения оплатить Директ со скидкой заведомо недобросовестны.

---

\* При некоторых условиях в редких случаях системе не удастся спрогнозировать недельный бюджет.

Требования к рекламным материалам

Рекламные материалы, размещаемые через Яндекс.Директ должны в первую очередь соответствовать общим требованиям к рекламным материалам, размещаемым на Яндексе ([http://legal.yandex.ru/general\\_adv\\_rules/](http://legal.yandex.ru/general_adv_rules/)).

Объявление должно состоять из:

заголовка (до 33 символов, включая пробелы и знаки препинания);  
собственно текста (до 75 символов, включая пробелы и знаки препинания);  
ссылки на сайт (URL страницы сайта, содержащей рекламное предложение) и/или контактных данных.

Заголовок и текст объявления должны быть написаны без опечаток грамотным русским языком. Выбранные ключевые слова и рубрики каталога должны соответствовать теме объявления.

Используемый в объявлении URL должен соответствовать теме объявления и стабильно работать. Не допускается использование IP-адреса вместо названия домена.

Контактные данные должны быть указаны корректно и без опечаток, а предложение, озвученное по телефону, должно соответствовать тексту объявления.

В заголовке и тексте объявления не допускается:

использование разрядки (написание слов с пробелами между б у к в а м и) и использование слов, набранных заглавными буквами. Исключение составляют аббревиатуры, состоящие из начальных букв сокращенных слов («СССР», «ГОСТ» и т.п.); указание номеров телефонов, e-mail адресов, почтовых адресов, номеров icq или других систем онлайн-пейджинга;  
использование html-кода.

В Яндекс.Директе не допускается реклама, которая может вводить потребителей в заблуждение относительно потребительских свойств, качества и количества товаров,

содержать ложные сведения или некорректные сравнения с товарами других производителей/продавцов (это явно запрещено в законе «О рекламе»).

Если в рекламе используются превосходные степени имен прилагательных и/или содержатся сведения о преимуществах рекламируемого товара перед находящимися в обороте товарами, то рекламная ссылка должна вести на страницу сайта рекламодателя, где эта информация подтверждена соответствующими исследованиями (заключениями и т.п.) третьих лиц.

Не допускается реклама, ведущая на сайт, основной целью которого является показ рекламы. К таковым, например, относятся сайты, содержащие страницы, где более половины площади первого экрана занимают рекламные блоки. Рекомендуем также ознакомиться с документом «Чем отличается качественный сайт от некачественного с точки зрения Яндекса?» (<http://webmaster.yandex.ru/faq.xml?id=985118>).

Создание большого числа одинаковых объявлений (с идентичным текстом, сайтом, ключевыми фразами), в том числе и в разных кампаниях, рассматривается как попытка спама. Такие объявления будут отклоняться как на этапе модерации, так и после начала работы кампаний.

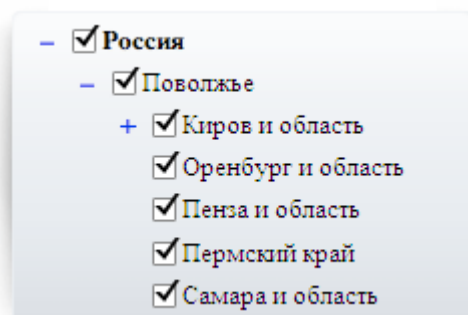
Обратите внимание:

Компания Яндекс оставляет за собой право отклонить любой рекламный материал при постмодерации без объяснения причин.

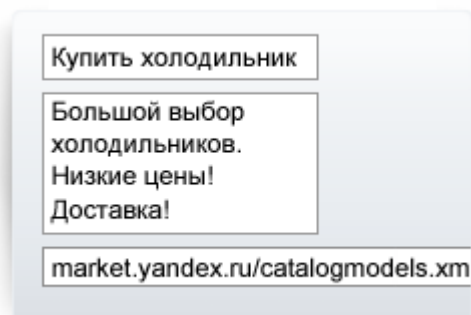
Как разместить объявление

Самостоятельно создать объявление в лёгком интерфейсе Яндекс.Директа можно всего за несколько шагов:

1. Укажите регионы, в которых хотите показывать объявление.



2. Напишите заголовок и текст объявления. Укажите адрес сайта или создайте страницу с контактной информацией — виртуальную визитку.



3. Подберите ключевые слова — ваше объявление будет показано пользователям, которые введут их на поиске Яндекса.

купить холодильник с доставкой, купить холодильник в Кирове, холодильники samsung, двухкамерные холодильники, недорогие холодильники, где купить холодильник

4. Выберите стратегию показа объявлений. Рекламный бюджет будет расходоваться в соответствии с выбранной стратегией.



- ☒ заданное число посетителей
- ☐ быстрое привлечение
- ☐ распределенный бюджет

5. Настройте время показа объявления, укажите свою контактную информацию и отправьте объявление на проверку.

|     |                                     |                                     |                                     |                                     |                                     |                                     |                                     |                                     |
|-----|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Пн. | <input checked="" type="checkbox"/> | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   |
| Вт. | <input checked="" type="checkbox"/> | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   |
| Ср. | <input checked="" type="checkbox"/> | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   |
| Чт. | <input checked="" type="checkbox"/> | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   |
| Пт. | <input checked="" type="checkbox"/> | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   | +                                   |
|     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
|     |                                     | 9                                   | 10                                  | 11                                  | 12                                  | 13                                  | 14                                  | 15                                  |

6. После того как ваше объявление будет проверено модератором, вы сможете оплатить заказ.

директ

Показы объявлений начнутся после оплаты заказа Оплатить

### Мои объявления

№ заказа: 2357217

Оплачено: 0.00 руб.

Остаток: 0.00 руб. [оплатить](#)

Посетителей: -

7. Вы можете оплатить рекомендуемую системой сумму или указать свою (от 300 рублей). Рекомендуемый бюджет рассчитан на неделю, меньшая сумма может быть расходована быстрее.

8. Выберите способ оплаты.

9. Зарегистрируйте плательщика.

10. Выпишите счет и оплатите его.

После поступления денег на счёт рекламной кампании, объявление появится на поиске Яндекса и на страницах сайтов — участников Рекламной сети.

Чтобы разместить объявление, перейдите по ссылке «Дать объявление».

Как рекламироваться без сайта

Рекламировать товары и услуги в Яндекс.Директе можно и без сайта. Для этого при создании объявления заполните форму с контактными данными "**Адрес и телефон**", и страничка с информацией о вашей компании (виртуальная визитка) будет создана автоматически.

Так выглядит виртуальная визитка:

Россия, Москва

 [Напечатать](#)

Телефон:

**+7 (800) 333-96-39, доб. 2480**

**Служба поддержки**

Время работы:

**пн-пт 09:00—19:00**

**сб-вс 10:00—19:00**

E-mail:

**[support@direct.yandex.ru](mailto:support@direct.yandex.ru)**

Адрес:

**Льва Толстого, д. 16**

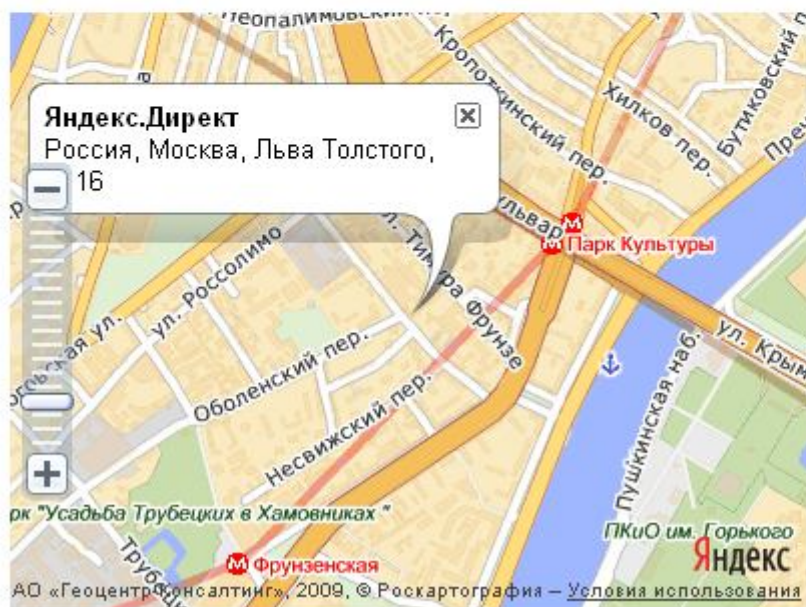
## Яндекс.Директ

Яндекс.Директ - это инструмент для размещения контекстных рекламных объявлений на страницах Яндекса и на сайтах-участниках его рекламной сети.

### [Реклама на Яндексе!](#)

Разместите объявление о своих товарах и услугах. От 30 копеек за клиента!

[direct.yandex.ru](http://direct.yandex.ru)



[Перейти на Яндекс.Карты](#)

В контактные данные к объявлению входят:

контактный номер телефона (код страны, код города или код мобильного оператора, номер телефона);

название компании и/или ФИО контактного лица;

дни недели и время работы, в течение которого на указанный телефон можно звонить по объявлению;

адрес (страна, город, улица, дом);

адрес электронной почты;

дополнительное текстовое поле для подробного описания товаров и/или услуг.

Чтобы обозначить время работы как "круглосуточное", укажите время работы с 00:00 до 24:00.

Если вы укажете свой адрес и он будет найден на Яндекс.Картах с точностью до улицы, то на странице с контактной информацией появится фрагмент карты и ссылка на этот фрагмент в Яндекс.Картах. Если указанный вами адрес будет точно найден на Яндекс.Картах, то во фрагменте карты будет поставлена метка в нужной точке. Вы всегда можете уточнить местоположение организации, установив ручную точку на карте.

Контактные данные должны быть указаны корректно и без опечаток, а предложение, озвученное по телефону, должно соответствовать тексту объявления. Эта информация проверяется модераторами.

При показе объявлений в **мобильных устройствах** ссылка на виртуальную визитку будет называться "Позвонить". При переходе по данной ссылке пользователь увидит только номер контактного телефона, указанный в контактной информации.

Какую стратегию показов выбрать

Перед тем, как сделать заказ, необходимо выбрать стратегию показов объявлений.

Стратегии позволяют максимально эффективно управлять рекламой в соответствии с вашими целями.

### **Заданное число посетителей**

Стратегия «**Заданное число посетителей**» дает вам возможность с определенной скоростью и за определенный бюджет получить желаемое количество посетителей сайта (виртуальной визитки).

Укажите в настройках стратегии количество посетителей, которых вы планируете привлечь с помощью рекламных объявлений. На основе этого параметра, а также выбранных вами ключевых фраз и региона показа будет спрогнозирован бюджет и срок размещения. Далее можно выбрать один из двух вариантов действий:

- Изменить сумму рекламного бюджета (тогда в интерфейсе вы увидите примерный срок, за который система приведет заданное число посетителей при вашем бюджете);
- Указать желаемый срок привлечения посетителей (в этом случае система перерассчитает для вас примерный бюджет рекламной кампании).

Блок выбора стратегии "Заданное число посетителей" в интерфейсе:

СТРАТЕГИИ: **Заданное число посетителей** [Быстрое привлечение посетителей](#) [Распределенный недельный бюджет](#)

Сколько новых посетителей Вы намерены привлечь?  посетителей

Какой бюджет Вы рассчитываете на это выделить?  руб.

Срок привлечения посетителей: [14 дней](#)

 Указанное кол-во дней - это прогноз на основании параметров конкурирующих объявлений. На практике срок привлечения посетителей может отличаться.

Значения прогнозируемых параметров стратегии (количество посетителей, бюджет и срок привлечения посетителей) зависят от выбранных вами ключевых фраз и региона показа, а также друг от друга. При изменении хотя бы одного из них, система автоматически скорректирует допустимые значения остальных. Например, если при заданном ограничении бюджета невозможно привлечь посетителей быстрее, чем за месяц, вы не сможете установить в качестве срока день или неделю.

### **Обратите внимание:**

- Срок привлечения посетителей прогнозируется с учетом многих факторов, включая конкуренцию по вашей тематике и сезонность спроса. При изменении любого из этих факторов прогнозируемый срок может изменяться как в большую, так и в меньшую сторону.
- Минимум посетителей, который можно задать в настройках — 100. Максимально возможное число посетителей определяется исходя из характеристик рекламной кампании и бюджета;
- Возможные сроки привлечения посетителей варьируются от 1 дня до 1 года. При этом минимальный и максимальный доступный срок привлечения посетителей для конкретной рекламной кампании зависят от ее параметров;
- Если в кампании используются низкочастотные ключевые слова и прогнозируемое число посетителей в пересчете на неделю менее 100, возможно отклонение в числе привлеченных посетителей от указанного рекламодателем значения, но не более чем в 2 раза;

- Если вы заплатили за рекламу больше указанного ограничения бюджета, то после привлечения заданного количества пользователей на заданный бюджет показы продолжатся с теми же настройками.
- Если вы заплатили за рекламу меньше указанного ограничения бюджета, система привлечет столько посетителей, на сколько хватает средств.

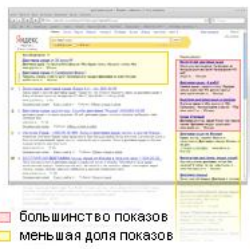
### Быстрое привлечение посетителей

Стратегия "**Быстрое привлечение посетителей**" позволяет распределить бюджет так, чтобы на ваш сайт или виртуальную визитку за как можно более короткий срок пришло как можно больше посетителей. При этом система будет учитывать, на каких позициях вы хотели бы показывать объявление — на лучших, средних или ниже средних. Рядом с названием позиции вы увидите прогнозируемое количество посетителей, которых можно будет привлечь по вашим ключевым фразам в неделю. При создании счета за рекламу система автоматически рассчитает бюджет, которого по прогнозам будет достаточно для показов на выбранных позициях в течение недели. На этапе выставления счета будет спрогнозирован срок, на который хватит указанных средств.

На лучших позициях объявление видит больше пользователей, и скорость их привлечения заметно выше. Но следует учитывать, что такая реклама стоит дороже, и бюджет будет расходоваться значительно быстрее.

Блок выбора стратегии "**Быстрое привлечение клиентов**" в интерфейсе:

СТРАТЕГИИ: [Заданное число посетителей](#) **[Быстрое привлечение посетителей](#)** [Распределенный недельный бюджет](#)



| Где показывать объявление                               | посетителей в неделю* |
|---------------------------------------------------------|-----------------------|
| <input type="radio"/> лучшие позиции                    | 350                   |
| <input checked="" type="radio"/> <b>средние позиции</b> | 290                   |
| <input type="radio"/> позиции ниже средних              | 60                    |

**Средние позиции** - позиции справа с 1-ой по 4-ую.  
Мы будем стремиться показывать объявления на этих позициях, при недостатке бюджета показы могут идти ниже 4-ой позиции.

■ большинство показов  
■ меньшая доля показов

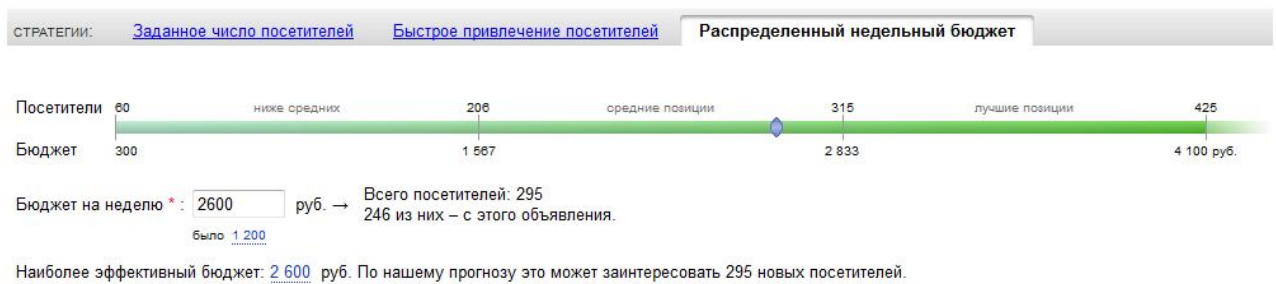
\* Прогноз. На практике количество посетителей может заметно отличаться.

### Обратите внимание:

- В Яндекс.Директе позиции, на которых показываются объявления, зависят от цены за переход по нему. Система не сможет обеспечить вам размещение на лучших или средних позициях, если бюджета хватает только на позиции, расположенные ниже средних.
- Если вы внесли средства в соответствии с рекомендациями системы при выставлении счета, система будет стремиться обеспечить вашим объявлениям показы на выбранных позициях.
- Как прогноз бюджета, так и прогноз скорости привлечения посетителей основаны на статистических данных и не могут учесть некоторые факторы, например, привлекательность текста вашего объявления или сезонность спроса на ваше предложение. Итоговый бюджет и количество посетителей могут отличаться от прогнозируемых.

### Распределенный недельный бюджет

При выборе стратегии "**Распределенный недельный бюджет**" оплаченные средства равномерно распределяются на календарную неделю. Система регулирует показы ваших объявлений таким образом, чтобы на протяжении недели за указанную сумму вы получили максимальное количество посетителей сайта (или виртуальной визитки). В случае нехватки средств показы автоматически приостанавливаются в течение дня. Блок выбора стратегии "**Распределенный недельный бюджет**" в интерфейсе:



Если объявлений несколько, то недельный бюджет распределяется по всем активным объявлениям.

В блоке, где вы указываете размер недельного бюджета, показана зависимость количества посетителей вашего сайта (или виртуальной визитки) от выбранной суммы.

#### Обратите внимание:

- Рекомендуемый бюджет и ожидаемое количество посетителей, которые вы видите при настройке стратегии, являются результатом прогноза, основанного на анализе ключевых фраз во всех ваших активных объявлениях. Итоговый бюджет и количество посетителей могут отличаться от прогнозируемых.

Как оплатить объявление

Способы оплаты

Ответы на часто задаваемые вопросы

Способы оплаты

После того, как объявления будут допущены модератором у вас появится возможность оплатить заказ, нажав на кнопку «Оплатить» на странице "Мои объявления".

Минимальный объем заказа составляет 300 рублей. При переводе на тариф «Беззаботный» размер минимального объема заказа может быть иным.

Оплатить заказ можно несколькими способами:

#### Для физических лиц:

**квитанция Сбербанка или любого другого банка, принимающего платежи от физических лиц** – вы можете выписать квитанцию в системе Яндекс.Директ и распечатать ее для последующей оплаты. Яндекс.Директ сформирует для вас квитанцию, в которой уже будут проставлены реквизиты Яндекса и указана сумма. Оплатить квитанцию можно в любом отделении Сбербанка или любого другого банка, принимающего платежи от физических лиц.

**кредитная карта** – к оплате принимаются карточки Visa и EuroCard/MasterCard (Cirrus Maestro и VISA Electron не принимаются). Платеж происходит в режиме реального времени и зачисляется в течение 15 минут.

**платежная система Яндекс.Деньги или Интернет.Деньги** – платеж происходит в режиме реального времени и зачисляется в течение 15 минут. При этом оплата Яндекс.Деньгами должна осуществляться с кошелька, находящегося на том же логине, что и кампания в Яндекс.Директе.

оплата через систему **WebMoney**.

с помощью **Кода платежа** – наличными в офисах наших партнеров, через терминалы оплаты и банкоматы или с помощью системинтернет-банкинга. Платеж зачисляется в течение 15 минут.

#### Для юридических лиц:

**счет для безналичной оплаты** – вы можете сформировать счет в Директе, в котором уже будут проставлены реквизиты Яндекса и указана сумма. Далее вы можете распечатать счет и оплатить его как вам удобно (например, через систему клиент-банк). Оригиналы бухгалтерских документов (счет, счет-фактура и акт) высылаются по почте ежемесячно.

Клиенты из Украины могут выставить счет в Директе самостоятельно. В случае возникновения вопросов Вы можете обратиться в офис Яндекс.Украина по адресу support@direct.yandex.ua или по телефонам +380 (48) 737-44-10, +380 (44) 586-41-48, 0 800 60-48-61 (звонок со стационарных телефонов в Украине бесплатный), и они

возьмут на себя выставление счета в гривнах, а также подготовку необходимых налоговых документов.

Все счета и квитанции выставляются нашей системой автоматически.

**Обратите внимание:**

Если вы можете предоставить платежное поручение с отметкой банка о проведении платежа или квитанцию с штрих-кодом Яндекса об оплате с отметкой банка, **обязательно** пришлите нам копию на адрес [info@balance.yandex.ru](mailto:info@balance.yandex.ru) или по факсу +7 (495) 739-2332, чтобы мы могли сразу же начать показ ваших объявлений. В противном случае ваша кампания начнется тогда, когда платеж будет зачислен банком на наш расчетный счет, что может занять до 2 недель, поскольку это зависит от работы банка. Если вы пользуетесь системой клиент-банк и не можете предоставить копию платежного поручения с пометкой банка, зачисление вашего платежа займет до двух рабочих дней. В случае, если Вы оплатили счет в гривнах, показ ваших объявлений начнется только после того, как платеж будет зачислен банком на наш расчетный счет. Для того, чтобы оперативно узнавать о том, что средства на рекламной кампании подошли к концу или закончились, можно подписаться на SMS-оповещения на мобильный телефон.   
наверх

Ответы на часто задаваемые вопросы

**1. Где можно посмотреть историю платежей?**

Историю платежей по заказу можно посмотреть, перейдя по ссылке "Платежи" со страницы "Мои объявления" в Директе.

**2. Какой порядок предоставления документов при оплате по безналичному расчету?**

Рассылка документов происходит в среднем 1 раз в неделю и осуществляется в два этапа: Рассылка документов по электронной почте, на адрес указанный вами в карточке плательщика.

Рассылка оригиналов документов Почтой России.

Оригиналы документов высылаются по почте в соответствии со следующим порядком: оригиналы бухгалтерских документов юридическим лицам высылаются традиционной почтой ежемесячно;

если оказание услуг по произведенной оплате заняло больше месяца, то по состоянию на последнее число месяца будет сформирован и отправлен клиенту промежуточный акт на сумму услуг, оказанных в рамках произведенной оплаты за отчетный месяц. Если оказание услуг в рамках одной оплаты растягивается на несколько месяцев, то такие промежуточные акты будут формироваться в конце каждого месяца. По окончании оказания услуг будет выписан акт на остаток суммы, услуги по которой оказывались со времени оформления последнего промежуточного акта; документы по каждой оплате или доплате формируются отдельно.

Пожалуйста, при заполнении информации о плательщике не забудьте указать правильный почтовый адрес, на него будут отсылаться бухгалтерские документы.

Электронная рассылка полностью соответствует бумажной, поэтому вы можете воспользоваться электронной версией документа для формирования бухгалтерской отчетности.

**Обратите внимание:**

В разделе «Платежи» вашего аккаунта теперь можно получить информацию по истории Акт о оказании услуг, которые были сформированы по вашим платежам. А именно: номер и дата Акта;

номер и дата счета-фактуры;

сумму Акта и счета-фактуры в рублях РФ.

Для того, чтобы увидеть эту информацию, достаточно кликнуть на интересующий номер счета в закладке «Счета».

**3. На выставленных счетах нет печати. Что нам делать?**

Пожалуйста, напишите по адресу support@direct.yandex.ru и укажите номер счета. Мы отправим вам счет с печатью по факсу или почтой.

В любом случае, счет с печатью будет находиться среди бухгалтерских документов высылаемых ежемесячно.

#### 4. Бухгалтерия не принимает счет на оплату без договора. Что нам делать?

Один из способов заключения договора – это акцепт оферты (публичного предложения, содержащего условия договора). Каждый клиент Яндекс.Директа, отправляя кампанию в первый раз на модерацию, соглашается с условиям договора оферты. Таким образом, оплата счета производится на его основании и договор считается заключенным согласно Гражданскому кодексу РФ. Таким образом, оплаченный счет + текст оферты являются юридически заключенным договором. Никаких печатей и подписей на оферте быть не должно.

Заключение договора возможно для компаний, обслуживаемых по тарифу «Беззаботный».

Как сделать объявление эффективным

Пишите правильные тексты и заголовки объявлений

Подбирайте верные ключевые фразы

Уточняйте регион, в котором будет показано объявление

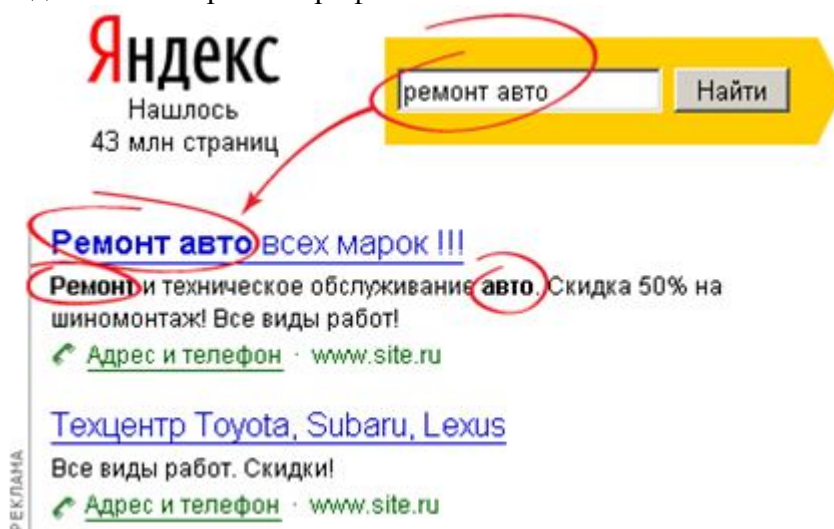
Указывайте правильное время показов объявлений

Пишите правильные тексты и заголовки объявлений

В контекстной рекламе главное – дать конкретный ответ на запрос пользователя. Именно поэтому основная рекомендация по написанию текста и заголовка в Директе – это конкретность.

— **Пишите правильные заголовки.** Дайте пользователю понять, что он нашел именно то, что искал сразу в заголовке объявления. Если пользователь спросил про «ремонт холодильников», то лучше уже в заголовке сообщить, что вы предлагаете именно эту услугу.

— **Используйте ключевые фразы в тексте или заголовке объявления.** Если в объявлении содержатся ключевые фразы, то при показе на поиске Яндекса они выделяются жирным шрифтом.



— **Создавайте несколько конкретных объявлений вместо одного общего.** Составьте отдельное объявление для каждого предложения, которое вы готовы сделать потенциальным клиентам.

— **Укажите цену.** Если пользователь будет заранее знать цену, то клик по объявлению будет означать его заинтересованность в приобретении товара по этой цене.

— **Назовите регион.** Если ваше предложение предназначено для конкретного региона, назовите его в заголовке или тексте объявления.

— **Расскажите об уникальности вашего предложения:** доставке, бонусах, скидках.

— **Используйте слова-стимуляторы, чтобы привлечь внимание пользователей именно к вашему объявлению.** Например: дарим, распродажа, скидка, подарок, недорогой, доступный, дешевый, бесплатно, сниженные, торопитесь.

— **Не используйте название компании в заголовке или тексте объявления.** Это не станет привлекающим фактором, если только вы не рекламируете легко узнаваемые бренды.

— **Побуждайте!** Объявления, которые содержат фразу, побуждающую к какому-либо действию (например, "Звоните!" или "Кликайте!"), зачастую более эффективны, потому что стимулируют ответную реакцию.

— **Укажите ссылку на страницу, которая содержит сведения о товарах и услугах, о которых идет речь в объявлении.** Если пользователи не смогут сразу найти то, о чем шла речь в объявлении, они покинут ваш сайт. Удостоверьтесь, что на странице, на которую придут посетители, содержится информация о рекламируемых товарах или услугах.

Обратите, пожалуйста, внимание на наши требования к рекламным материалам.  
наверх

Подбирайте верные ключевые фразы

Представьте себе человека, который хочет найти ваш товар или услугу. Какие запросы он может ввести в строку поиска Яндекс? Ваша задача — предугадать запросы потенциального клиента и перечислить их в разделе "Подберите ключевые фразы".

Человек, который введет одну из этих фраз на поиске Яндекса, увидит ваше объявление.

Подберите ключевые фразы

вызов такси, такси цены, грузовое такси

↓

вызов такси срочно

Найти

Пользователь увидит объявление, только если его запрос будет целиком содержать ключевую фразу. Например, если вы задали словосочетание "дизайн интерьера", то ваше объявление будет показано по запросам "журнал интерьер и дизайн", "заказать дизайн интерьера", но не будет показано по запросу "дизайн" или по запросу "интерьер".

1. Ключевые фразы перечисляются через запятую:

Подберите ключевые фразы

евроокна, окна цены, пластиковые окна

2. Используйте ключевую фразу "купить продукт" вместо "продукт". Человек, который набирает в поиске "купить продукт" с большей вероятностью готов к покупке, чем тот,

кто ищет по слову "продукт". Такую же функцию выполняют и другие слова — мы называем их "продающие добавки".

3. Посмотрите, какие фразы можно позаимствовать из текстов на вашем сайте или из прайс-листа вашей компании.

4. Подумайте о более конкретных формулировках ваших ключевых слов. Например, к фразе "отдых в Турции" можно добавить названия турецких курортов: "Анталия", "Кемер", "Мармарис", "Бодрум" и др.

5. Подумайте о профессионализмах и сленге. Например: "наружная реклама" и "наружка".

6. Используйте разные части речи. Например, к словосочетанию "магазин мебели" можно добавить "мебельный магазин".

7. Пишите названия иностранных марок и латиницей, и кириллицей (например, Toyota и Тойота).

8. Помните о распространенных опечатках. Известно, что 14% всех запросов на Яндексe содержат ошибки. Например, слово "риэлтор" запрашивается в 27 вариантах его написания: "риэлтер", "риелтор", "риелтер", "резлтор", "реелтор", "риэлтр" и др.

9. Учитывайте возможные ошибки в написании сложносоставных слов. Например, слово "металлочерепица" нередко пишут в два слова "металло черепица".

10. Воспользуйтесь сервисом "Подбор слов".

В форме на странице подбора слов введите ключевое слово или словосочетание, обозначающее ваш товар или услугу, и нажмите на кнопку "подобрать". Результат подбора будет поделен на два столбца.

Что искали со словом

**«ремонт»** — 708893 показов в месяц

| Слова                             | Показов в месяц |
|-----------------------------------|-----------------|
| <a href="#">ремонт</a>            | 708893          |
| <a href="#">ремонт квартир</a>    | 47296           |
| <a href="#">ремонт машин</a>      | 18367           |
| <a href="#">ремонт телефонов</a>  | 18207           |
| <a href="#">школа ремонта</a>     | 16250           |
| <a href="#">ремонт автомобиля</a> | 15077           |

Что еще искали люди, искавшие **«ремонт»**:

| Слова                                    | Показов в месяц |
|------------------------------------------|-----------------|
| <a href="#">*ремонт квартир</a>          | 47296           |
| <a href="#">*ремонт офисов</a>           | 4858            |
| <a href="#">дизайн квартир</a>           | 31086           |
| <a href="#">дизайн проект квартиры</a>   | 2316            |
| <a href="#">дизайн проект</a>            | 13158           |
| <a href="#">интерьер квартиры</a>        | 13814           |
| <a href="#">*ремонт квартир в москве</a> | 1081            |

В левом столбце будут приведены запросы пользователей, включающие заданное вами ключевое слово или словосочетание. Цифра рядом с каждым запросом представляет собой предварительный прогноз числа показов в месяц, которое вы получите, выбрав этот запрос в качестве ключевой фразы. Таким образом, словосочетания из левого столбца покажут вам сферу интересов, для обозначения которой пользователи употребляли ваше ключевое слово. Если эти интересы заметно шире, чем специфика вашего бизнеса, ограничьте свою аудиторию одним из двух способов:

Добавьте в ключевое словосочетание еще одно или несколько слов. Например, если вы занимаетесь ремонтом автомобилей ВАЗ, вам может быть лучше показывать объявление не по словосочетанию "ремонт автомобилей", а по словосочетанию "ремонт автомобилей ВАЗ".

Используйте минус-слова, чтобы объявление не показывалось в ответ на похожие, но не подходящие по смыслу запросы. В ответ на поисковые запросы, содержащие ваши минус-слова, объявление показано не будет.

## Подберите ключевые фразы

дизайн интерьера — реферат — книга, дизайн офиса, дизайн помещения

В правом столбце будут приведены другие запросы, которые делали люди, искавшие введенное вами слово. Они упорядочены по убыванию показателя "близости" с исходным запросом. Добавив ключевые слова из этого списка, вы сможете показать свое объявление большему числу пользователей.

11. Для уточнения ключевой фразы или слова используйте восклицательный знак (!), плюс (+) или кавычки (").

(!) Если перед ключевым словом поставить восклицательный знак, объявление будет показано при любом запросе, содержащем это слово точно в указанном написании. Например, если необходимо, чтобы объявление показывалось по запросам "Сони", "магнитола Сони", "наушники Сони" и др., но не показывалось по запросу "Соня значение имени" (название компании совпадает с родительным падежом женского имени), то в качестве ключевой фразы укажите "!Сони".

(+) Яндекс не учитывает служебные части речи (предлоги, союзы) и местоимения в составе ключевой фразы — как и их присутствие в поисковом запросе. Например, ключевая фраза "все для дома" идентична ключевому слову "дом", так как служебные слова "все" (местоимение) и "для" (предлог) игнорируются системой. Если поставить плюсы перед этими словами ("+все +для дома"), то они будут учитываться: ваше объявление будет показываться только тем пользователям, в запросе которых присутствуют все три слова из вашей ключевой фразы.

("") Если вы поставите ключевую фразу в кавычки, рекламное объявление увидят только те пользователи, которые введут запрос, содержащий именно эту фразу. Например, если ключевую фразу "медицинские анализы" взять в кавычки, то в ответ на запрос "медицинские анализы на дому" объявление показано не будет.

наверх

Уточняйте регион, в котором будет показано объявление

Если вы укажете географический регион, в котором находятся ваши клиенты, то объявления будут показаны только в этом регионе. Это позволит повысить эффективность вашей рекламной кампании, исключить нежелательных посетителей вашего сайта, а так же снизить затраты на рекламную кампанию.

Чтобы указать регион показа ваших объявлений перейдите по ссылке "Уточнить регион" при создании/редактировании объявления и укажите любое количество нужных вам географических регионов, в которых вы хотите показывать свои объявления.

наверх

Указывайте правильное время показов объявлений

Если специфика рекламируемых услуг такова, что показывать рекламу лучше в определенные дни недели и время суток, то можно указать время работы заказа. Тогда система автоматически будет включать показы всех объявлений в указанный промежуток времени и приостанавливать их в остальное время.

Указывая время показов, вы ограничиваете время **показов** рекламы, а не кликов. Имейте это ввиду при работе с настройками заказа, поскольку клик по объявлению может произойти заметно позже показа.

Вы можете изменить время показа в настройках заказа — отметить те дни недели и часы, в которые должны показываться объявления, выбрать часовой пояс, задать расписание показа в праздничные дни.

## Выбор часового пояса..

При настройке расписания показов можно указать подходящий часовой пояс. По умолчанию указано московское время. Для изменения часового пояса кликните на название пояса в поле «Время».

### Обратите внимание:

- a. Для выбора доступны часовые пояса, которые отличаются от времени по Гринвичу только на целое число часов.
- b. Данные в статистике рассчитаны по **московскому** времени.
- c. Переход на летнее время будет осуществляться автоматически для соответствующих часовых поясов.

### Управление показами в праздничные дни и в рабочие выходные.

В Яндекс.Директе вы можете управлять показами объявлений в праздничные дни. С помощью данной опции вы можете либо задать расписание показов, либо запретить показы ваших объявлений в праздники. Для этого в настройках времени показа поставьте галочку «Учитывать праздничные дни» и укажите подходящий график показов.

## Расписание показа объявлений

Вы можете настроить время, когда будут показываться рекламные объявления. Обратите внимание: показы должны быть разрешены не менее 40 часов в рабочее время в неделю.

Показывать объявления: ☒ Круглосуточно ☐ В рабочее время ☒ Другое...

| Пн. | Вт. | Ср. | Чт. | Пт. | Сб. | Вс. | 00 <sup>01</sup> | 01 <sup>02</sup> | 02 <sup>03</sup> | 03 <sup>04</sup> | 04 <sup>05</sup> | 05 <sup>06</sup> | 06 <sup>07</sup> | 07 <sup>08</sup> | 08 <sup>09</sup> | 09 <sup>10</sup> | 10 <sup>11</sup> | 11 <sup>12</sup> | 12 <sup>13</sup> | 13 <sup>14</sup> | 14 <sup>15</sup> | 15 <sup>16</sup> | 16 <sup>17</sup> | 17 <sup>18</sup> | 18 <sup>19</sup> | 19 <sup>20</sup> | 20 <sup>21</sup> | 21 <sup>22</sup> | 22 <sup>23</sup> | 23 <sup>00</sup> |
|-----|-----|-----|-----|-----|-----|-----|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| -   | -   | -   | -   | -   | -   | -   | -                | -                | -                | -                | -                | -                | -                | -                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | -                | -                | -                |
| -   | -   | -   | -   | -   | -   | -   | -                | -                | -                | -                | -                | -                | -                | -                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | -                | -                | -                |
| -   | -   | -   | -   | -   | -   | -   | -                | -                | -                | -                | -                | -                | -                | -                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | -                | -                | -                |
| -   | -   | -   | -   | -   | -   | -   | -                | -                | -                | -                | -                | -                | -                | -                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | -                | -                | -                |
| -   | -   | -   | -   | -   | -   | -   | -                | -                | -                | -                | -                | -                | -                | -                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | -                | -                | -                |
| -   | -   | -   | -   | -   | -   | -   | -                | -                | -                | -                | -                | -                | -                | -                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | +                | -                | -                | -                |
| -   | -   | -   | -   | -   | -   | -   | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                | -                |

Время: [Москва](#) Всего часов в рабочие дни: 60

☒ Учитывать рабочие выходные<sup>2</sup>

☒ Учитывать праздничные дни<sup>2</sup>: ☒ показывать с 08:00 до 20:00 ☐ не показывать

Для случаев, когда в праздники рабочий день переносится на субботу или воскресенье, в интерфейсе предусмотрена опция «Учитывать рабочие выходные». Благодаря ей можно настроить показы рекламы в рабочий выходной по расписанию соответствующего ему рабочего дня.

Чтобы включить опцию, отметьте галочкой пункт "Учитывать рабочие выходные" в настройках временного таргетинга. Для новых рекламодателей учет рабочих выходных включается по умолчанию.

## Обратите внимание

В настройках учитываются официальные праздничные дни и переносы выходных дней только по законодательству РФ. Праздничным днем считается непосредственно праздничная дата, а также будний день, который стал официальным выходным за счет переноса праздничного дня.

### Важно:

Ограничивать время показов рекламных объявлений можно до 40 часов в неделю в рабочие дни.

Более подробную информацию вы можете получить в брошюре "Как составить эффективное объявление. Советы от Яндекса", (PDF, 1 291 Кб).

## Почему объявления не показываются

Объявление еще не прошло модерацию

Объявление отклонено модератором

Заказ не оплачен

В браузере закешировалась старая информация

Ключевая фраза отключена за низкую эффективность

Объявление не попало в гарантированные показы

Включено ограничение показов объявления по времени суток и дням недели (временному таргетингу)

Включено ограничение показов объявления по регионам показа (геотаргетингу)

Объявление еще не прошло активизацию

Объявление еще не прошло модерацию

Рекламные материалы – текст, ключевые слова, ссылка с объявления (если есть), модерируются обычно в течение двух часов. В ночное время модерация не производится.

Модерация контактной информации из виртуальной визитки проводится в промежутки времени, указанные в контактных данных, поэтому время модерации контактной информации может длиться более суток, в отличие от модерации рекламных материалов заказа.

В некоторых случаях объявление может быть допущено к показам автоматически. Это означает, что объявление начнет показываться практически сразу после создания или внесения изменений, однако будет дополнительно проверено модератором на предмет соответствия требованиям к рекламным материалам позже и может быть отклонено по результатам этой проверки. Модерация автоматически принятых объявлений проводится в течение трех суток.

О решении модератора вас известят письмом. Кроме того, решение будет отражено в статусе вашего заказа на странице "Мои объявления".

**Режим работы модерации:** 7 дней в неделю с 9 до 24 часов.

наверх

Объявление отклонено модератором

Текст объявления и/или рекламируемые товар/услуга могут быть отклонены по следующим причинам:

**текст не согласован или содержит опечатки** — исправьте опечатки в тексте или заголовке объявления;

**в заголовке, тексте, а также в URL объявления не допускается набор слов заглавными буквами** — в соответствии с требованиями к рекламным материалам исключениями являются зарегистрированные товарные знаки и аббревиатуры, состоящие из начальных букв сокращенных слов («СССР», «ГОСТ» и т.п.);

**в заголовке и тексте не допускается использование разрядки** — написание слов с пробелами между б у к в а м и;

**содержание рекламируемого сайта или страницы не соответствует теме/тексту объявления** — например, если реклама содержит информацию о скидке или рекламной акции, то рекламная ссылка должна вести на ту страницу сайта рекламодателя, где эта скидка явно указана;

**страница сайта, на которую ссылается рекламное объявление, не открывается или сайт не доступен** — в момент модерации сайт был не доступен или на нем отсутствовала указанная вами страница;

**в заголовке, тексте объявления не должны упоминаться телефоны, e-mail, номера icq или почтовые адреса** — правилами рекламной политики Яндекса запрещено указывать контактные данные в объявлении, но вы можете разместить контактные данные на специальной странице, которую мы покажем посетителям при клике на ваше объявление.

**объявление не соответствует действующему законодательству** — рекламируемая вами услуга или товар не соответствуют действующему законодательству Российской Федерации, и мы не можем принять к размещению ваши рекламные материалы;

**Закон "О Рекламе" при дистанционном способе продажи не соблюден** — согласно статье 8 Закона "О рекламе" в рекламе товаров при дистанционном способе продажи должны быть указаны: название организации, адрес, ОГРН — для юридических лиц; фамилия, имя, отчество, ОГРН — для индивидуальных предпринимателей.

**авторское право** — если рекламируемый товар относится к объектам интеллектуальной собственности, необходимо предоставить документ, подтверждающий право его использования (как правило, договор). Пришлите соответствующие документы с указанием номера рекламной кампании по электронному адресу [support@direct.yandex.ru](mailto:support@direct.yandex.ru) или по факсу +7 (495) 739-2332.

**текст содержит неподтвержденную информацию, ложные сведения или некорректные сравнения** — это запрещено законом "О Рекламе", поэтому необходимо исправить текст объявления;

**услуги/товар подлежит лицензированию/сертификации** — не предоставлены необходимые документы. Ознакомьтесь порядком предоставления документов..

**объявление не соответствует рекламной политике Яндекса** — объект рекламирования противоречит рекламной политике Яндекса, и мы не можем принять к размещению ваши рекламные материалы.

**сайт не соответствует требованиям** — требования к сайту рекламодателя изложены в пунктах 7, 8, 10 на странице: <http://advertising.yandex.ru/trebovaniya1.xml>

**не соответствует требованиям к рекламе новостных ресурсов** — если реклама новостных ресурсов содержит информацию о новостной статье, опубликованной на сайте рекламодателя, то рекламное объявление должно вести на ту страницу сайта, где опубликована данная статья. Текст рекламного объявления должен соответствовать по смыслу заголовку и содержанию новостной статьи. При этом из текста рекламного объявления должно быть понятно, что оно рекламирует источник, на котором можно ознакомиться с новостной статьей. Подробнее пункт 4.1 на странице: <http://advertising.yandex.ru/trebovaniya1.xml>

Ключевые слова объявления могут быть отклонены по следующим причинам:

**ключевые слова/словосочетания должны разделяться запятыми** — разделите ключевые слова запятыми;

**ключевые слова не соответствуют теме объявления** — задача контекстной рекламы точно соответствовать запросу пользователя, поэтому контекстная реклама не принимается по общим запросам, например: продажа, покупка, аренда, заказ, цена, доставка, скидки, прокат, бесплатно, дисконт, скупка и т. д. Оцените, насколько выбранные ключевые слова/словосочетания соответствуют вашему рекламному предложению и исключите слова, имеющие наименьшее соответствие;

**нет пробела в минус-словах** — добавьте пробелы между минус-словами;

Контактная информация может быть отклонена по следующим причинам:

**в контактной информации или подробном описании содержатся опечатки** — исправьте опечатки в написании страны/города/адресе/названии компании, ФИО контактного лица или подробном описании товара/услуги;

**неправильно введен номер телефона или добавочный номер/контактное лицо** — номер телефона или добавочный, который вы указали, принадлежит другому лицу/организации или не существует, исправьте номер телефона;

**номер телефона недоступен/не отвечает** — в указанное вами время работы дозвониться по телефону с нескольких попыток не удалось, исправьте время работы или измените номер телефона на доступный;

**неверная контактная информация** — внимательно проверьте введенную контактную информацию и исправьте ее.

**Обратите внимание:**

В случае, если вы уверены, что объявления были отклонены по ошибке, или вам не до конца понятна причина отклонения, вы можете написать письмо на адрес [support@direct.yandex.ru](mailto:support@direct.yandex.ru) с указанием номера заказа и обоснованием вашей позиции.

наверх

Заказ не оплачен

Показы объявлений начинаются только после зачисления денежных средств на заказ.

Подробнее узнать о способах оплаты и сроках зачисления платежей можно [здесь](#).

наверх

В браузере закешировалась старая информация

Для обновления информации нажмите клавиши **ctrl+F5**.

наверх

Ключевая фраза отключена за низкую эффективность

При показателе эффективности (CTR) ниже 0,5% ключевая фраза отключается, поскольку она не совсем соответствует теме объявления. **CTR**— отношение числа кликов на объявление к числу его показов, измеряется в процентах. Можно сказать, что это КПД (коэффициент полезного действия) объявления, определяющий эффективность его работы.

На странице заказа отключенные фразы подсвечены бледно-серым цветом.

Отключенные фразы обратно не включаются. Но Вы можете вместо отключенной фразы сделать несколько более точных фраз, включающих уточняющие слова. Чтобы улучшить эффективность рекламы, вы можете воспользоваться нашими советами в разделе [Как составить эффективное рекламное объявление?](#).

наверх

Объявление не попало в гарантированные показы

Проверьте показы вашего объявления на странице просмотра всех объявлений Яндекс.Директа по ключевым фразам.

наверх

Включено ограничение показов объявления по времени суток и дням недели (временному таргетингу)

Проверьте настройки вашего заказа и убедитесь, что текущее время или день недели соответствуют заданному времени показа объявлений.

наверх

Включено ограничение показов объявления по регионам показа (геотаргетингу)

В нашей базе может отсутствовать IP-адрес вашего компьютера или он принадлежит другому географическому региону. В этом случае вы не видите своего объявления, хотя оно показывается другим посетителям. Вы можете прислать свой IP-адрес на адрес [support@direct.yandex.ru](mailto:support@direct.yandex.ru), чтобы мы могли точно сказать в чем причина. Узнать ваш IP адрес вы можете [здесь](#).

наверх

Объявление еще не прошло активизацию

Между включением заказа и началом показа объявлений может пройти до 40 минут, в периоды особенно высокой нагрузки — не более 3 часов.

Почему у ключевой фразы появились дополнительные слова

Если ваша ключевая фраза работает неэффективно, то есть, при большом количестве показов, происходит недостаточное количество переходов по объявлению, она может быть автоматически уточнена. Система автоматического уточнения основана на статистике поисковых запросов пользователей. В результате действия системы к слову или словосочетанию могут быть добавлены **минус-слова** (слова, по запросам с которыми рекламное объявление показываться не будет) или **плюс-слова** так, чтобы ключевые фразы стали как можно ближе по смыслу к поисковому запросу потенциального клиента. Уточненные фразы помечаются специальным значком. Добавленные минус-слова выделяются серым цветом. В результате уточнения ключевых фраз работа объявления

становится более эффективной, то есть приводит больше целевых пользователей. Кроме этого, избавление от ненужных запросов позволяет в большинстве случаев снизить рекламный бюджет.

Например, по словосочетанию "минеральные воды" могут рекламироваться и доставка минеральной воды, и лечение в санаториях курорта Кавминводы. Имеющиеся данные позволяют в первом случае исключить показы по запросам про минеральные воды, содержащие такие слова, как "кавказские", "город", "аэропорт", а во втором — запросы со словами "офис", "рынок", "питьевая". Таким образом, рекламодатели избегают не только отключения ключевых слов за низкую эффективность, но и ценовой конкуренции по запросам, которые не являются целевыми для каждого из них.

#### **Обратите внимание:**

Поскольку получаемые после уточнения фразы составлены автоматически и включаются без предварительной проверки модератором, они могут содержать бренды конкурентов, ненормативную лексику и т.п., в зависимости от того, что содержалось в пользовательских запросах, по которым показывалось объявление. Такие фразы могут быть отклонены при постмодерации.

#### **Правила показа**

Показ Рекламных объявлений (далее по тексту также - объявления) на Рекламных местах сопровождается пометкой: "Яндекс.Директ", или "Директ", или "Реклама".

Объявления, размещенные через Яндекс.Директ, могут быть показаны:

на страницах результатов поиска (<http://www.yandex.xx/>) Яндекса по выбранному вами запросу — до 12 объявлений,

на страницах сайтов-участников Рекламной сети Яндекса

(<http://direct.yandex.ru/help/?id=990409>) — до 9 объявлений на тематических страницах и

до 12 объявлений на страницах результатов поиска по сайту,

на страницах рубрик Каталога Яндекса (<http://catalog.yandex.xx/>) и на страницах результатов поиска по Каталогу — до 12 объявлений,

на страницах результатов поиска по Яндекс.Картам (<http://maps.yandex.xx/>) — до 2 объявлений,

на страницах результатов поиска по блогам (<http://blogs.yandex.xx/>) — до 5 объявлений,

на страницах результатов поиска по Яндекс.Маркету (<http://market.yandex.ru/>) — одно объявление,

на страницах просмотра всех объявлений (<http://direct.yandex.xx/search>) Яндекс.Директа по ключевым фразам — все объявления, конкурирующие между собой.

На страницах сервисов Яндекса, оптимизированных для мобильных устройств, рекламные объявления могут не показываться либо максимальное их число может быть менее предусмотренного выше для соответствующего рекламного места. Место показа объявлений на страницах сервисов Яндекса, оптимизированных для мобильных устройств, может отличаться от мест показа объявлений, предусмотренных в настоящих Правилах показа.

При отборе объявлений для показа на ВСЕХ позициях результатов поиска учитывается ключевая фраза (без стоп-слов (<http://direct.yandex.ru/help/?id=990397#990402>)),

коэффициент качества, кликабельность и цена за клик, указанная рекламодателем.

При этом при отборе рекламных объявлений для показа на всех позициях результатов поиска при расчете показателя CTR учитываются только показы и клики на Страницах результатов основного поиска Яндекса. Показы и клики на страницах результатов других сервисов Яндекса (Видео, Картинки и т.д.), на других поисковых и тематических площадках никак на него не влияют.

При отборе рекламных объявлений для показа на страницах рубрик Каталога, при расчете показателя CTR учитываются только показы и клики на страницах соответствующей рубрики Каталога.

На страницах с результатами поиска Яндекса, поиска по КATALOGу Яндекса и в рубриках КATALOGа показывается до трех объявлений в Спецразмещении. Спецразмещение – это выигрышное рекламное место с высокой кликабельностью, расположенное над результатами поиска или над основным содержанием страницы в Яндекс.Каталоге.

The screenshot shows the top of a Yandex search results page. At the top is the search bar with the text 'поиск' and 'поиск'. Below the search bar, a dashed box highlights a 'Спецразмещение' (Special Placement) advertisement. The ad is titled 'Купи слова на Яндексе!' (Buy words on Yandex!) and includes the text 'Яндекс.Директ - контекстная реклама, которая приводит покупателей!' (Yandex.Direct - contextual advertising, which brings buyers!). Below this, there is a green icon and the text 'Адрес и телефон · advertising.yandex.ru'. To the left of the ad, the word 'реклама' (advertisement) is written vertically. Below the ad, there are several search results, each with a title, URL, and a brief description.

Остальные объявления показываются справа от результатов поиска.

The screenshot shows the top of a Yandex search results page. At the top is the search bar with the text 'поиск' and 'поиск'. Below the search bar, a dashed box highlights a 'Спецразмещение' (Special Placement) advertisement. The ad is titled 'Купи слова на Яндексе!' (Buy words on Yandex!) and includes the text 'Яндекс.Директ - контекстная реклама, которая приводит покупателей!' (Yandex.Direct - contextual advertising, which brings buyers!). Below this, there is a green icon and the text 'Адрес и телефон · advertising.yandex.ru'. To the left of the ad, the word 'реклама' (advertisement) is written vertically. Below the ad, there are several search results, each with a title, URL, and a brief description.

Одновременно на Страницах результатов поиска Яндекса СПРАВА может быть показано: до 4-х объявлений статически, то есть гарантированно при каждом поисковом запросе, точно совпадающем с указанной рекламодателем ключевой фразой (без стоп-слов) – это Гарантированные показы. до 5-ти объявлений, не попавших в Гарантированные показы, динамически (то есть, чередуясь) на позициях под Гарантированными показами – это Динамические показы. Общее количество объявлений, ротирующихся на этих позициях, практически не ограничено.

спецразмещение



1-ое место

гарантированные  
показы

медийный  
контекстный баннер

динамические  
показы

Объявления, могут не показываться на Страницах результатов поиска Яндекса, поскольку по некоторым запросам они могут вытесняться медийным контекстным баннером (<http://advertising.yandex.ru/media/context/>) и/или товарными предложениями Яндекс.Маркета (<http://market.yandex.ru/>).

Объявления с наилучшими показателями попадают в Спецразмещение и на Гарантированные показы. Остальные объявления попадают в Динамические показы.

Спецразмещение, в отличие от гарантированных показов, имеет порог входа, не зависящий от наличия конкурентов. Поэтому вполне нормальной будет ситуация, когда показывается много объявлений Яндекс.Директа, но ни одно из них не попадает в Спецразмещение. С другой стороны, объявления с высокой кликабельностью часто могут попадать в Спецразмещение по цене ниже цены за клик, установленной для показа на первом месте справа.

Объявления в Спецразмещении и в Гарантированных показах располагаются в порядке убывания цены за клик.

В Динамических показах частота показа зависит от количества ротирующихся по запросу объявлений, от их кликабельности и цены за клик. При показе эти объявления располагаются в порядке убывания величины, зависящей от кликабельности и цены за клик.

Рекламные объявления на страницах сервиса поиск Яндекса, оптимизированного для мобильных устройств, располагаются в порядке убывания произведения кликабельности на цену за Клик.

Отбор объявлений для показа и их показ на тематических площадках Рекламной сети Яндекса (<http://direct.yandex.ru/help/?id=990409>) осуществляется Яндексом автоматически на основании соответствия тематики объявлений интересам пользователя. При выборе в параметрах рекламной кампании критерия показа Рекламных объявлений «Независимое размещение на разных типах площадок», Рекламодатель может установить цену за Клик для показа на тематических страницах сайтов партнеров РСЯ. Значение «Охвата аудитории», отображаемое в клиентском интерфейсе при установке Рекламодателем цены за Клик для показа рекламного объявления на тематических страницах сайтов партнеров РСЯ, носит информативный характер и означает ориентировочное количество показов рекламного объявления в % отношении от количества показов рекламных материалов по данному ключевому слову на тематических страницах сайтов партнеров РСЯ.

Объявления на страницах рубрик Каталога, показываются согласно выбранным Рекламодателем рубрикам. Отбор объявлений для показа осуществляется Яндексом автоматически с учетом произведения кликабельности объявления на страницах соответствующей рубрики Каталога на цену за клик.

Отбор Рекламных объявлений для показа на страницах просмотра всех объявлений (<http://direct.yandex.xx/search>) Яндекс.Директа и их распределение по страницам осуществляется в порядке уменьшения произведения кликабельности и цены за Клик. При этом порядковое место Рекламного объявления на странице определяется в соответствии с установленной ценой на Клик, т.е. отобранные Рекламные объявления размещаются на каждой странице в порядке уменьшения цены за Клик.

Показ Рекламного объявления на страницах результатов поиска по Яндекс.Картам может сопровождаться отображением на карте специальной метки, с привязкой к точке местности, соответствующей адресу, указанному Рекламодателем на контактной странице для данного Рекламного объявления. При обращении пользователей к метке показывается блок «Развернутая метка», внутри которого содержится Рекламное объявление и/или предоставленная Рекламодателем контактная информация.

Обратите внимание:

1. Если CTR (по статистике Яндекса) Рекламного объявления по какому-либо ключевому слову/словосочетанию, указанному Рекламодателем для этого Рекламного объявления или для соответствующей рубрики Каталога, окажется ниже порогового значения, установленного в настоящем пункте, Яндекс вправе приостановить или прекратить размещение (показ) этого Рекламного объявления по такому ключевому слову на соответствующем рекламном месте.

| Рекламное место                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Пороговое значение CTR                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| Страницы:<br>результатов поиска Яндекс;<br>результатов просмотра всех объявлений<br>( <a href="http://direct.yandex.xx/search">http://direct.yandex.xx/search</a> ) Яндекс.Директа;<br>результатов поиска по Каталогу;<br>результатов поиска по Яндекс.Картам<br>( <a href="http://maps.yandex.xx/">http://maps.yandex.xx/</a> );<br>результатов поиска по блогам<br>( <a href="http://blogs.yandex.xx/">http://blogs.yandex.xx/</a> );<br>результатов поиска по Яндекс.Маркету<br>( <a href="http://market.yandex.ru/">http://market.yandex.ru/</a> );<br>результатов поиска на страницах сайтов партнеров РСЯ. | 0,5% на страницах результатов поиска Яндекса ( <a href="http://www.yandex.xx/">http://www.yandex.xx/</a> ) |
| Страницы рубрик Каталога                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0,5% на страницах соответствующей рубрики сервиса Каталог                                                  |

CTR означает соотношение количества Кликов к количеству показов рекламного объявления.

2. При пересечении ключевых фраз и/или рубрик Каталога в различных объявлениях, находящихся в одной рекламной кампании, ставки по этим ключевым словам не будут поднимать друг другу цену за клик. Для показа из них будет выбрано только одно, у которого произведение кликабельности на цену за клик является наилучшим.

3. При пересечении ключевых фраз и/или рубрик Каталога в объявлениях, ведущих на страницы одного и того же сайта, при показе будет выбрано только одно объявление, у которого произведение кликабельности на цену за клик является наилучшим.

4. В процессе работы рекламных кампаний отслеживается:

А) Коэффициент качества, влияющий на минимальную цену за клик, действующую на поиске. Коэффициент качества зависит от большого количества показателей и рассчитывается в режиме реального времени. В число основных показателей входят: общая релевантность объявления запросу, статистические данные по объявлению, релевантность страницы сайта, на которую ведет ссылка из объявления, и иные параметры.

Б) Параметр эффективности, влияющий на минимальную цену за клик, действующую на поиске. Значение этого параметра зависит от большого количества показателей, которые пересчитываются в режиме реального времени. В число основных показателей входят статистические данные (показы, CTR, количество дней активности кампаний) агрегированные по рекламируемому домену или номеру телефона (если он указан в "виртуальной визитке"). При ставке, ниже минимальной действующей на поиске, показы объявлений будут идти только на сайтах-участниках РСЯ и на странице "Все объявления".

4.1. Для целей расчета показателя кликабельности по рекламному объявлению учитываются данные за последние 28 дней, в течение которых осуществлялся показ данного рекламного объявления, либо используется прогнозируемый показатель кликабельности (при отсутствии достаточных данных статистики для его расчета за указанный выше период).

5. При размещении рекламы в системе Яндекс.Директ Яндекс вправе по своему усмотрению признавать рекламные объявления сходными. Сходными признаются рекламные объявления (одного и того же или разных рекламодателей), которые одновременно:

имеют идентичный или аналогичный объект рекламирования: одни и те же или сходные в значительной степени товары (работы, услуги), предлагаемые к продаже в одном регионе; размещаются по одинаковым или пересекающимся ключевым словам (словосочетаниям, рубрикам каталога);

содержат ссылки на один и тот же или сходные сайты и/или совпадение указанных и/или фактических контактных реквизитов рекламодателей (поставщиков товаров).

При этом сайты признаются сходными при совпадении значительной части ассортимента предлагаемых на сайтах товаров (работ, услуг), реализуемых в одном регионе, и/или наличии других признаков, позволяющих предположить, что предлагаются одни и те же товары одним и тем же (аффилированными, имеющими тесную взаимосвязь) поставщиком, вне зависимости от незначительного различия ассортимента, разного дизайна, структурной организации сайтов, отличия цен, а также от принадлежности (администрирования) сайтов разным(-и) лицам(-и).

При размещении рекламных объявлений по соответствующему ключевому слову (словосочетанию) из группы объявлений, признанных сходными, пользователю показывается только одно рекламное объявление, у которого произведение кликабельности на цену за клик является наилучшим. Все остальные рекламные объявления такой группы признаются дублирующими и не показываются (не размещаются). Назначенная рекламодателями для дублирующих объявлений цена за клик не учитывается (в том числе при определении места показа иных рекламных объявлений) в течение всего времени приостановки показа таких объявлений.

В указанных случаях Яндекс принимает решение о признании объявлений сходными с учетом оценки имеющейся в его распоряжении информации по собственному усмотрению.

5.1. В случае, если Ссылки у рекламного объявления Яндекс.Директа и товарного предложения Яндекс.Маркета ведут на один и тот же информационный ресурс (сайт)/домен в сети Интернет, либо на сайты, признанные сходными в указанном в п. 5 порядке, на страницах результатов поиска Яндекса будет показано только рекламное

объявление или только товарное предложение, у которого произведение кликабельности на цену за Клик будет наилучшим.

6. Особенности работы рекламной кампании при изменении ее состояния:

6.1. Обработка действий, совершенных Рекламодателем через клиентский интерфейс (или через API), в отношении управления Рекламной кампанией, осуществляется в течение следующего срока:

изменение цены за Клик от 5 до 60 минут (средний срок обработки данного действия составляет 10 минут);

остановка Рекламной кампании от 10 до 60 минут (средний срок обработки данного действия составляет 30 минут);

иные действия от 5 до 180 минут (средний срок обработки данных действий составляет 40 минут).

6.2. При изменении Рекламодателем цены за Клик, Клики оплачиваются по старой цене до момента обработки данного действия Яндексом;

6.3. При остановке показа Рекламной кампании, услуги перестанут оказываться с момента обработки данного действия Яндексом. При этом в данный период Клики оплачиваются в соответствии с установленной Рекламодателем ценой;

6.4. Информация об обработке действий, связанных с изменением статуса Рекламной кампании, указывается в клиентском интерфейсе.

7. При отображении Рекламных объявлений на мобильных устройствах в Рекламном объявлении могут присутствовать Ссылка на сайт и Ссылка на контактную страницу ("Позвонить"). При обращении пользователя к Ссылке на контактную страницу, пользователю может быть показан только номер контактного телефона, указанный для данного рекламного объявления в виртуальной визитке.

8. Цена за Клик.

Цена за Клик может быть установлена Рекламодателем в следующих пределах: от 0.01 у.е. до 50 у.е., с учетом правил, изложенных ниже:

8.1. В случаях, когда цена на Клик, установленная Рекламодателем, превышает сумму остатка денежных средств на Рекламной кампании или сумму установленного Рекламодателем ограничения стоимости услуг, считается, что цена на Клик равна сумме остатка денежных средств на Рекламной кампании или сумме установленного Рекламодателем ограничения стоимости услуг (ее неизрасходованному остатку) соответственно, если иное ограничение не применимо согласно п. 8.2.

8.2. Максимальный размер цены на Клик для Рекламной кампании, который может быть установлен Рекламодателем в рамках Рекламной кампании, определяется Яндексом в момент наступления следующих событий: а) увеличения остатка денежных средств на Рекламной кампании; б) выставления Счета с отсрочкой платежа; в) изменения Рекламодателем цены за Клик; г) изменения остатка денежных средств на Рекламной кампании, произошедшего в результате действий, совершенных Рекламодателем через клиентский интерфейс; д) изменения активных ключевых слов; е) установления/изменения ограничения стоимости услуг, произошедшего в результате действий, совершенных Рекламодателем через клиентский интерфейс.

Максимальный размер цены за Клик, который может быть установлен Рекламодателем в рамках Рекламной кампании, рассчитывается в следующем порядке:

Остаток денежных средств на Рекламной кампании  $\times 2$  / суммарный прогноз Кликов, где суммарный прогноз Кликов =  $\sum$  прогнозов Кликов по всем ключевым словам Рекламной кампании.

По каждому ключевому слову Прогноз Кликов рассчитывается, как статистика ключевого слова /  $1440 \times \text{CTR}$ , где статистика ключевого слова = количеству запросов (<http://wordstat.yandex.ru/>), содержащих ключевое слово в месяц.

$\text{CTR} = 0.6\%$ , если цена за Клик установлена Рекламодателем менее 0,21 у.е.,  $\text{CTR} = 2,56\%$ , если цена на Клик установлена Рекламодателем больше (или равна) 0.21 у.е. и меньше

1.07 у.е., CTR = 4.12%, если цена на Клик установлена Рекламодателем больше (или равна) 1.07 у.е. и меньше 5.09 у.е., CTR = 12.73%, если цена на Клик установлена Рекламодателем больше (или равна) 5.09 у.е.

8.2.1. В случае, если рассчитанный максимальный размер цены за Клик для Рекламной кампании меньше 0.05 у.е., то максимальный размер цены за Клик для Рекламной кампании будет установлен в размере 0.05 у.е.

8.2.2. Во всех случаях, когда цена за Клик, установленная Рекламодателем превышает максимальный размер цены за Клик для Рекламной кампании, применяется максимальный размер цены на Клик.

Значение максимального размера цены за Клик для Рекламной кампании будет показано в клиентском интерфейсе после окончания обработки действий, указанных в первом абзаце п. 8.2.

9. Объявления медицинской или табачной тематики сопровождаются предусмотренными ФЗ «О рекламе» предупреждениями стандартного вида, а именно:

«Курение вредит вашему здоровью» — для рекламы табака и табачных изделий;  
«Есть противопоказания. Посоветуйтесь с врачом» — для рекламы лекарственных средств, медицинских услуг, в том числе методов лечения, медицинской техники;  
«Есть противопоказания. Посоветуйтесь с врачом. Возможен вред здоровью или бесплодие» — для рекламы медицинских услуг по искусственному прерыванию беременности.

Сопровождение объявлений предупреждениями указанного вида производится путем автоматического определения тематики объявлений и/или через службу поддержки support@direct.yandex.ru. Контроль за надлежащим определением тематики объявления и обязательным наличием в объявлениях соответствующей тематики необходимого предупреждения обязуется обеспечивать Рекламодаделец.

Объявления с предупреждениями временно не будут показываться на страницах Яндекс.Каталога

10. Если для объявления были заполнены дополнительные ссылки, они отображаются при показе объявления на позиции первого Спецразмещения. При показе объявления на других позициях/страницах, даже если дополнительные ссылки были заполнены, объявление показывается без них.

11. При использовании рекламодателем опции "Показы по дополнительным релевантным фразам", Рекламодаделец обязуется обеспечить контроль за соответствием добавленных ключевых слов/словосочетаний требованиям законодательства, в том числе требованиям Федерального закона от 13.03.2006 года №38-ФЗ "О рекламе", Федерального закона от 26.07.2006 года №135-ФЗ "О конкуренции". Рекламодаделец может установить ограничение стоимости услуг при показах по дополнительным релевантным фразам в процентах по отношению к общей стоимости услуг. Указанное ограничение применяется к стоимости услуг, оказываемых не менее, чем в течение одной недели с даты установления такого ограничения.

12. Рекламодаделец может установить ограничение стоимости услуг при показах в Рекламной сети Яндекса в процентах по отношению к общей стоимости услуг. Указанное ограничение применяется к стоимости услуг, оказываемых не менее, чем в течение одной недели с даты установления такого ограничения.

13. Яндекс вправе периодически в целях повышения качества системы Яндекс.Директ и оказываемых услуг, а также разработки новых предложений для рекламодателей проводить эксперименты (не более одного эксперимента одновременно с продолжительностью не более 2 недель), в рамках которых показ Рекламных объявлений на персональных компьютерах и мобильных устройствах отдельных выбранных системой в случайном порядке Пользователей (количество таких Пользователей – не более 10% всей аудитории), может осуществляться на Рекламных местах с изменением условий отбора и порядка расположения Рекламных объявлений на Рекламных местах,

предусмотренных настоящими Правилами. При проведении экспериментов Яндекс не гарантирует, что все без исключения Рекламные объявления будут показаны с соблюдением порядка расположения Рекламных объявлений на Рекламных местах и условий отбора, определенных настоящими Правилами. При этом проведение экспериментов не влияет на установленную Рекламодателем в соответствии с п. 8 настоящих Правил Цену за Клик и не влечет ее изменение. Оплата Услуг осуществляется в соответствии с Ценой за Клик, определенной в порядке, предусмотренном п. 8 настоящих Правил.

Информация о проведении таких экспериментов доступна Рекламодателю при обращении в службу поддержки путем направления запроса по электронной почте на [support@direct.yandex.ru](mailto:support@direct.yandex.ru) с указанием причины обращения и номера рекламной кампании. По указанному запросу Рекламодателю предоставляется информация о предстоящих экспериментах, проведение которых начинается не позднее 7 (семи) календарных дней с даты направления Рекламодателем запроса, а также информация о проводимых в момент получения запроса экспериментах.

14. Рекламодаделец может отказаться от показа Рекламных объявлений:

на страницах сайтов-участников Рекламной сети Яндекса

(<http://direct.yandex.ru/help/?id=990409>), добавив адрес сайта в список запрещенных

площадок в параметрах рекламной кампании или установив запрет показа для соответствующего сайта Рекламной сети Яндекса на странице статистики по рекламным площадкам (<http://direct.yandex.ru/help/?id=990418>).

в Спецразмещении, установив в параметрах рекламной кампании критерий показа Рекламных объявлений «Показ справа от результатов поиска».

на страницах результатов поиска (<http://www.yandex.xx/>) всех сервисов Яндекса, а также

на страницах результатов поиска сайтов партнеров РСЯ, установив в параметрах рекламной кампании критерий показа Рекламных объявлений «Независимое размещение на разных типах площадок» с опцией «Показы на поиске отключены».

15. Для целей настоящего документа обозначение «xx», используемое в доменных именах, указанных в настоящем документе, означает любую доменную зону верхнего или второго уровня, в которой Яндекс осуществляет администрирование домена с указанным именем.

---

1 у.е. = 30 руб.

Дата публикации 30.08.2012 г.

Предыдущая версия документа: [http://legal.yandex.ru/direct\\_display\\_rules/23072012/](http://legal.yandex.ru/direct_display_rules/23072012/).

Предыдущая версия документа: [http://legal.yandex.ru/direct\\_display\\_rules/26062012/](http://legal.yandex.ru/direct_display_rules/26062012/).

Предыдущая версия документа: [http://legal.yandex.ru/direct\\_display\\_rules/24052012/](http://legal.yandex.ru/direct_display_rules/24052012/).

Предыдущая версия документа: [http://legal.yandex.ru/direct\\_display\\_rules/15052012/](http://legal.yandex.ru/direct_display_rules/15052012/).

Предыдущая версия документа: [http://legal.yandex.ru/direct\\_display\\_rules/20092011/](http://legal.yandex.ru/direct_display_rules/20092011/).

Что такое Бегун?

О компании

Пресс-центр

Спецпроекты

Мероприятия

Исследования

Обучение

Технологии

Сотрудничество

Контакты

Вакансии

«Бегун» – это сервис интернет-рекламы с аукционным ценообразованием. «Бегун»

позволяет проводить в режиме «одного окна» рекламные кампании на сайтах-партнерах,

во всех крупнейших поисковых системах Рунета, а также во внешних рекламных сервисах.

| Презентация О компании «Бегун»ppt (4.6 Мб)

Для рекламодателей

«Бегун» предлагает линейку продуктов интернет-рекламы для решения любых маркетинговых задач. Сервис позволяет купить конкретный результат: переход на сайт, информирование уникального зрителя, звонок в офис. Стоимость результата определяется самим рекламодателем. Чтобы зарегистрироваться и начать кампанию, потребуется не более 10 минут. Минимальная цена контракта: 140 рублей (без НДС).

Для сайтов

«Бегун» является крупнейшей рекламной сетью в Рунете. Подключиться к ней может любой сайт, обладающий качественной аудиторией. Это наиболее удобный и выгодный способ конвертировать аудиторию в деньги.

Для агентств

Независимо от масштабов и специализации любое агентство может начать предоставлять своим клиентам услуги рекламы в «Бегуне». Для этого достаточно пройти обучение и аккредитацию. Выгодные условия и востребованность контекстной рекламы обеспечат присутствие «Бегуна» в каждом медиаплане.

# **Лекция 5. Размещение сайта в сети. Выбор хостинга**

## Правильный выбор хостинга

### Первичный выбор хостинга:

A1) **Выбор веб-студии.** Чтобы понять, какой хостинг вам нужен, необходимо для начала составить перечень основных требований к нему. Грамотный разработчик сайтов всегда предоставит вам такой список, чем облегчит ваше будущее общение с сотрудниками компании, предоставляющей хостинг. Некоторые разработчики не только **порекомендуют** какую-нибудь хостинговую компанию, но и **предложат свой собственный сервер** для размещения вашего сайта.

В первом случае — доверяй, но проверяй. Иногда студия преследует свои собственные интересы, получая комиссионное вознаграждение от хостеров за каждого приведенного клиента.

Во втором случае, когда компания-разработчик предлагает собственный хостинг, надо сразу понимать, что такая услуга, возможно, **не сможет сравниться по качеству с профессиональной.**

Что делать, если вам нужен просто хостинг под **файловый архив**? - Обращаем внимание на доступное **место на диске.**

Вам нужен **интернет-магазин**? Многие хостеры сдают их в аренду.

Нужна **cms**?

*Система управления содержимым (контентом) (Content management system, CMS) — информационная система или программа, используемая для организации совокупности процессов: создания, редактирования и управления [контентом](#).*

Нужен **windows** или **unix** хостинг?

Возможно, вы хотите именно русский (или только американский) сервер (датацентр).

Вам может понадобится хостинг с flash communication server (Flash Media Server - FMS, сервером флеш от Macromedia для задач потокового видео, виде по запросу и др.)

Быть может, вам жизненно необходим "**хостинг без политической цензуры**"? Такого рода вопросы и составляют, собственно, начало поиска правильного хостинга.

A2) **По совету знакомых.** Если Вы решили не пользоваться советами веб-студии, то за помощью в выборе хостинговой компании лучше обращаться к независимым советчикам. Полезными могут оказаться рекомендации друзей или знакомых. По статистике, **порядка 30%** клиентов находят хостинговую компанию **по совету знакомых.**

### Изучение поставщиков услуг:

После первичных поисков вы оказались перед огромным выбором. Как же выбрать подходящую компанию? Для начала определитесь с типом услуг хостинга. Варианты хостинга различаются поддерживаемыми технологиями и порядком размещения сайта:

**По технологиям** — 1.**UNIX** (языки программирования php, perl) и 2.**WINDOWS** (язык программирования asp).

**По порядку размещения** — виртуальный 1.**shared-хостинг**, 2.**dedicated** (виртуальный выделенный сервер), 3.**co-location** (физический выделенный сервер).

Наиболее распространенная хостинговая услуга — виртуальный shared-хостинг на платформе UNIX.

**Виртуальный shared-хостинг** — это дисковое пространство на сервере хостинг-провайдера, выделенное под размещение файлов вашего сайта. На одном сервере с вашим сайтом будут размещены несколько сотен сайтов других клиентов. Для большинства сайтов этого вполне достаточно.

Но если у вас очень посещаемый ресурс, или занимающий много дискового пространства, или для вас критически важно качество работы — выбирайте выделенный сервер — dedicated и co-location.

Что делать дальше? Определите для себя набор существенно важных условий, например:

1. Ценовые ограничения — нужен дешевый хостинг или параметр цена-качество важен больше.
2. Принципиально ли наличие офиса в вашем городе?
3. Методы оплаты услуг — безналичный расчет, наличные, Сбербанк, кредитная карта, системы электронных платежей.
4. Наличие лицензии на предоставление услуг (лицензия на услуги телематической передачи данных).
5. Известность торговой марки провайдера и размер клиентской базы, как давно компания присутствует на рынке?

Из провайдеров, удовлетворяющих входным условиям, выберите 5-7 для дальнейшей селекции. Важным условием отбора может быть **количество доменных имен в зоне RU, поддерживаемых данным провайдером**. Этот параметр показывает относительное (НЕ точное!) количество клиентов у данного провайдера (чем больше доменных имен, тем больше клиентов).

Изучите сайты отобранных хостинг-провайдеров. На сайте должна быть представлена подробная информация о самом провайдере, тарифных планах, условиях и методах оплаты, описание оборудования и технической площадки, текст договора.

Зайдите на форум хостинг-провайдера, соберите контакты нескольких его клиентов, попробуйте с ними связаться и узнать их мнение о провайдере. Если на форуме только восторженные отзывы и нет критики или обсуждения технических вопросов, неполадок, значит провайдер, их удаляет и «рыльце у него в пушку». **Серьезные компании не стесняются информировать клиентов о существующих проблемах и ходе их решения.**

Внимательно **изучите текст договора**, там прописываются правила пользования сервисами провайдера, при нарушении которых ваш сайт может быть заблокирован без возврата уплаченных денег (рассылка спама, размещение запрещенной информации, превышение допустимого уровня

нагрузки на оборудование провайдера).

Выберите **подходящий по техническим параметрам тарифный план**. Планы различаются набором поддерживаемых технологий, количеством выделяемого пространства под сайт, почтовых ящиков, баз данных и сайтов, которые можно одновременно размещать.

Зарегистрируйтесь на бесплатный тестовый период и разместите на предоставленном временном адресе копию своего сайта. Многие хостеры, не умеющие предоставить хостинг на хорошем уровне безопасности, вводят большое количество технических ограничений на работу сайтов и скриптов (например, safe mode), которые сильно затрудняют корректную работу сайтов и жизнь разработчикам сайтов. Если **провайдер не предоставляет тестового периода** — это повод задуматься. Может ему **есть что скрывать?**

Если же работа вашего сайта в тестовом периоде вас удовлетворила, обратите внимание на службу поддержки хостинг-провайдера. **Служба поддержки должна работать круглосуточно, 7 дней в неделю**. Составьте для себя список из 5-10 вопросов, которые вы хотите задать службе поддержки (например, порядок и скорость регистрации доменных имен, порядок оплаты и предоставления документов, неясные вам пункты договора).

Служба поддержки **должна отвечать на электронные письма быстро** (в течение 12 часов, максимум 24) и корректно. Ответ должен быть вежливым и понятным неспециалисту, то есть вам. Обязательно должен быть телефон для более оперативной связи, некоторые провайдеры дополнительно предоставляют поддержку по ICQ.

### Панель управления учётной записью

Последний пункт для изучения — панель управления вашим аккаунтом (учетной записью). Если панели управления нет, вам придется по каждому поводу обращаться в службу поддержки.

Какие функции должны быть на панели управления? Технические настройки — смена паролей на доступ к сайту и панели управления, редактирование настроек сайта и баз данных, создание почтовых ящиков, заказ доменных имен, создание поддоменов. Статистика: количество занятого места, нагрузка на сервер, до какого числа оплачены услуги. Организационные настройки — выписка счетов на хостинг и доменные имена, список контактных лиц, которым разрешен доступ к сайту и настройкам, реквизиты плательщика.

Если все через все эти проверки с честью прошли 2-3 компании, а Вы все еще не знаете какую из них выбрать — **посетите офис провайдера**, пообщайтесь с директором.

## Дата-центр

---

**Дата-центр** (от [англ. data center](#)), или **центр (хранения и) обработки данных (ЦОД/ЦХОД)** — это специализированное здание для размещения ([хостинга](#)) [серверного](#) и [коммуникационного оборудования](#) и подключения абонентов к каналам сети [Интернет](#).

Дата-центр выполняет функции [обработки](#), [хранения](#) и [распространения](#) информации, как правило, в интересах корпоративных клиентов — он ориентирован на решение бизнес-задач путём предоставления информационных услуг. Консолидация [вычислительных ресурсов](#) и [средств хранения](#)

[данных](#) в ЦОД позволяет сократить совокупную стоимость владения ИТ-инфраструктурой за счёт возможности эффективного использования технических средств, например, перераспределения нагрузок, а также за счёт сокращения расходов на администрирование.

Дата-центры обычно расположены в пределах или в непосредственной близости от [узла связи](#) или [точки присутствия](#) какого-либо одного или нескольких [операторов связи](#). Качество и пропускная способность каналов влияют на уровень предоставляемых услуг, поскольку основным критерием оценки качества работы любого дата-центра является время доступности сервера ([аптайм](#)).

## История

История развития дата-центров начинается с огромных компьютерных комнат времен зарождения компьютерной индустрии. Тогда компьютерные системы были сложнее в управлении и требовали обеспечения особых условий для работы. Так как они занимали много места и требовали множества проводов для подключения различных компонентов, в компьютерных комнатах стали применять стандартные [серверные стойки](#), [фальшполы](#) и [кабельные каналы](#) (проложенные по потолку или под [фальшполом](#)). Кроме того, такие системы потребляли много энергии и нуждались в постоянном охлаждении, чтобы оборудование не перегревалось. Не менее важна была безопасность — оборудование весьма дорогостоящее и часто использовалось для военных нужд. Поэтому были разработаны основные конструкционные принципы по контролю доступа в [серверные](#).

В период бурного развития компьютерной индустрии и особенно в 1980-е компьютеры начинают использовать повсеместно, не особенно заботясь об эксплуатационных требованиях. Но с развитием ИТ-отрасли, компании начинают уделять все больше внимания контролю ИТ-ресурсов. С изобретением [архитектуры клиент-сервер](#) в 1990-е, микрокомпьютеры, сейчас называемые [серверами](#), стали занимать места в старых [серверных](#). Доступность недорогого сетевого оборудования вместе с новыми стандартами сетевых кабелей сделали возможным использование [иерархического проектирования](#), так серверы переехали в отдельные комнаты. Термин «дата-центр», в те времена применимый к специально спроектированным [серверным](#), начал набирать популярность и становился все более узнаваем.

Бум дата-центров приходится на период 1995—2000 годов. Компаниям было необходимо устойчивое и высокоскоростное соединение с Интернетом и бесперебойная работа оборудования, чтобы разворачивать системы и устанавливать своё присутствие в сети. Разместить оборудование, способное справиться с решением этих задач, было делом непосильным для большинства небольших компаний. Тогда и началось строительство отдельных больших помещений, способных обеспечить бизнес всем необходимым набором решений для размещения компьютерных систем и их эксплуатации. Стали развиваться новые технологии для решения вопросов масштаба и операционных требований столь крупных систем.

В наши дни проектирование и строительство дата-центров — хорошо изученная область.

Сформированы стандарты, устанавливающие требования для проектирования дата-центров. К примеру специалисты [Telecommunications Industry Association](#) внесли большой вклад в формирование

стандартов для дата-центров. Но все же осталось ещё много нерешенных задач в методах работы, а также строительстве дата-центров, не наносящих вреда окружающей среде и т. д.

Дата-центры требуют больших затрат как на этапе строительства, так и в процессе обслуживания, для поддержания работы на должном уровне. К примеру, дата-центр Amazon.com в Орегоне, площадь которого равна 10, 800 кв.м, оценивается в 100 млн \$.<sup>[1]</sup>

## Структура

---

Типичный дата-центр состоит из:

- информационной инфраструктуры, включающей в себя серверное оборудование и обеспечивающей основные функции дата-центра — обработку и хранение информации;
- телекоммуникационной инфраструктуры, обеспечивающей взаимосвязь элементов дата-центра, а также передачу данных между дата-центром и пользователями;
- инженерной инфраструктуры, обеспечивающей нормальное функционирование основных систем дата-центра.

Инженерная инфраструктура включает в себя: кондиционирование для поддержания температуры и уровня влажности в заданных параметрах; бесперебойное электроснабжение для автономной работы дата-центра в случаях отключения центральных источников электроэнергии; охранно-пожарную сигнализацию и [система газового пожаротушения](#); системы удаленного [IP контроля](#), [управления питанием](#) и контроля доступа.

Некоторые дата-центры предлагают клиентам дополнительные услуги по использованию оборудования по автоматическому уходу от различных видов [атак](#). Команды квалифицированных специалистов круглосуточно производят мониторинг всех серверов. Необходимо отметить, что услуги дата-центров сильно отличаются в цене и количестве услуг. Для обеспечения сохранности данных используются резервные системы копирования. Для предотвращения кражи данных, в дата-центрах используются различные системы ограничения физического доступа, системы видеонаблюдения. В корпоративных (ведомственных) дата-центрах обычно сосредоточено большинство серверов соответствующей организации. Оборудование крепится в специализированных [стойках](#) и [шкафах](#). Как правило, в дата-центр принимают для размещения лишь оборудование в стоечном исполнении, то есть в корпусах стандартных размеров, приспособленных для крепления в стойку. Компьютеры в корпусах настольного исполнения неудобны для дата-центров и размещаются в них редко.

## Классификация

---

### По соответствию требованиям стандартов

В ряде стран имеются [стандарты](#) на оборудование помещений дата-центров, позволяющие объективно оценить способность дата-центра обеспечить тот или иной уровень сервиса. Например, в США принят американский ([ANSI](#)) стандарт TIA-942, несущий в себе рекомендации по созданию дата-центров, и делящий дата-центры на типы по степени надёжности. Хотя в России пока нет такого стандарта, дата-

центры оснащаются согласно требованиям для сооружений связи, а также ориентируются на требования TIA-942 и используют дополнительную документацию [Uptime Institute](#) и ГОСТы серии 34.

Фактически, TIA-942 воспринимается во всем мире как единый стандарт для дата-центров, однако следует отметить что он достаточно давно не обновлялся и его достаточно сложно применить в условиях России. В то же время сейчас активно развивается стандарт BICSI 002 2010 Data Center Design and Implementation Best Practices, появившийся в 2010 и обновленный в 2011. По словам создателей стандарта "стандарт BICSI 002 2010, в создании которого участвовали более 150 экспертов, дополняет существующие стандарты TIA, CENELEC и ISO/IEC для центров обработки данных". Каждый из стандартов, как правило имеет свою внутреннюю классификацию дата-центров по совокупности их параметров.

## По размеру

- Крупные дата-центры имеют своё здание, специально сконструированное для обеспечения наилучших условий размещения. Обычно они имеют свои каналы связи, к которым подключают серверы.
- Средние дата-центры обычно арендуют площадку определённого размера и каналы определённой пропускной способности (обычно измеряется в Мбит/с).
- Малые дата-центры размещаются в малоприспособленных помещениях. Часто ими используется оборудование плохого качества, а также предоставляется самый минимум услуг.
- [Контейнерные дата-центры](#). Стойки с оборудованием размещаются в стандартных ISO контейнерах размером 20 и 40 футов. Имеют преимущества т.к. могут перевозиться автомобильным и железнодорожным транспортом.

## По надёжности

Основной показатель работы ЦОД — отказоустойчивость; также важна стоимость эксплуатации, показатели энергопотребления и регулирования температурного режима.

Например, стандарт TIA-942 предполагает четыре уровня надёжности дата-центров<sup>[2]</sup>:

- Tier 1 (N) — отказы оборудования или проведение ремонтных работ приводят к остановке работы всего дата-центра; в дата-центре отсутствуют [фальшполы](#), резервные источники электроснабжения и источники бесперебойного питания; инженерная инфраструктура не зарезервирована;
- Tier 2 (N+1) — имеется небольшой уровень резервирования; в дата-центре имеются фальшполы и резервные источники электроснабжения, однако проведение ремонтных работ также вызывает остановку работы дата-центра;
- Tier 3 (N+1) — имеется возможность проведения ремонтных работ (включая замену компонентов системы, добавление и удаление вышедшего из строя оборудования) без остановки работы дата-центра; инженерные системы однократно зарезервированы, имеется несколько каналов распределения электропитания и охлаждения, однако постоянно активен только один из них;

- Tier 4 (2(N+1)) — имеется возможность проведения любых работ без остановки работы дата-центра; инженерные системы двукратно зарезервированы, то есть продублированы как основная, так и дополнительная системы (например, бесперебойное питание представлено двумя ИБП, работающими по схеме N+1).

## По предназначению

Дата-центры по виду использования подразделяют на корпоративные, предназначенные для обслуживания конкретной компании, и коммерческие ([аутсорсинговые](#)), предоставляющие услуги всем желающим. Также разделяют провайдерозависимые и провайдеронезависимые дата-центры. Первые служат для обеспечения деятельности телекоммуникационных операторов, вторые могут использоваться разными компаниями в соответствии с их нуждами.

## Услуги дата-центров

---

- [Виртуальный хостинг](#). Крупные дата-центры обычно не предоставляют подобную массовую услугу из-за необходимости обеспечения техническо-консультационной поддержки.
- [Виртуальный сервер](#). Предоставление гарантированной и лимитированной части сервера (части всех ресурсов). Важная особенность данного вида хостинга — разделение сервера на несколько виртуальных независимых серверов, реализуемых программным способом.
- [Выделенный сервер](#). Дата-центр предоставляет клиенту в аренду сервер в различной конфигурации. Крупные дата-центры в основном специализируются именно на подобных типах услуг.
- [Colocation](#). Размещение сервера клиента на площадке дата-центра за определённую плату. Стоимость зависит от энергопотребления и тепловыделения размещаемого оборудования, пропускной способности подключаемого к оборудованию канала передачи данных, а также размера и веса стойки.
- Аренда [телекоммуникационных стоек](#). Передача клиенту стоек для монтажа собственного или клиентского оборудования. Формально это частный случай colocation, но с основным отличием в том, что арендаторы в основном юридические лица.
- Выделенная зона (Dedicated area). В некоторых случаях владельцы дата-центра выделяют часть технологических площадей для специальных клиентов, как правило, финансовых компаний, имеющих строгие внутренние нормы безопасности. В этом случае дата-центр предоставляет некую выделенную зону, обеспеченную каналами связи, электроснабжением, холодоснабжением и системами безопасности, а клиент сам создает свой дата-центр внутри этого пространства.

## Сетевая [инфраструктура](#)

---

Коммуникации дата-центра чаще всего базируются на сетях с использованием протокола [IP](#). Дата-центр содержит несколько [роутеров](#) и [свитчей](#), которые управляют [трафиком](#) между серверами и «внешним миром». Для надёжности дата-центр иногда подключен к интернету с помощью множества разных внешних каналов от разных [провайдеров](#).

Некоторые серверы в дата-центре служат для работы базовых интернет- и [интранет](#)-служб, которые используются внутри организации: почтовые сервера, [прокси-сервера](#), DNS-сервера и т. п.

Сетевой уровень безопасности поддерживают [межсетевые экраны](#), [VPN](#)-шлюзы, [IDS](#)-системы и т. д. Также используются системы мониторинга трафика и некоторых приложений.

## «Треш-дата-центры»

---

В последние годы на фоне [глобального кризиса](#), получили развитие проекты максимально бюджетных «треш-дата-центров» ([англ. trash](#) — мусор). Как правило, это большой склад либо ангар, минимально приспособленный для размещения стоек, который по классификации TIA-942 не всегда соответствует даже уровню Tier 1. Клиент обеспечивается негарантированным электропитанием, подключением к Интернет и местом для стойки. Если клиент желает иметь гарантированное электропитание, то должен поставить собственный ИБП в стойку. Охлаждение оборудования происходит за счёт естественного воздухообмена в помещении.

Достоинством данной услуги для клиента является крайне низкая цена, составляющая до 20—30 % от стоимости размещения сервера в дата-центрах уровня Tier 2 и Tier 3. Также в случае данного типа дата-центров можно говорить о некоторой экологичности, поскольку нет затрат на охлаждение помещения и оборудования (в случае же «обычного» дата-центра на это расходуется до 30 % от величины его общего энергопотребления).

## Дата-центры в России

**В России количество дата-центров постоянно растет: для этого переоборудуются существующие здания, а также строятся новые. В основном их местоположение ограничено крупными региональными центрами, такими как Москва и Санкт-Петербург. На их возникновение влияет наличие в том или ином месте централизованных [точек обмена трафиком](#). Точной количественной информации по распределению коммерческих дата-центров в мире не существует<sup>[3]</sup>. Однако судить о том, в каких регионах услуги дата-центров представлены наиболее широко, можно, например, с помощью независимого ресурса DataCenterMap, который был создан совладельцем одной из компаний, предоставляющих услуги хостинга, с целью облегчить поиск подходящего дата-центра потенциальным клиентам. Datacentermap пополняется данными при участии компаний — владельцев дата-центров, при этом вносимые данные в обязательном порядке проверяются владельцем ресурса.**

## Пожары в дата-центрах

---

- 27 марта 2010 года 21:00 по киевскому времени в дата-центре крупного украинского хостинг-провайдера в Одессе (ул. Дальницкая, 46) начался пожар. Система пожаротушения не сработала, а прибывшие на место пожарные не имели газовых средств огнетушения. В результате серверы пострадали как от воздействия огня, так и от воды<sup>[4]</sup>.
- 30 мая 2008 года в дата-центре «The Planet» в [Хьюстоне](#) по причине короткого замыкания возник пожар. Огонь не дошёл до серверов, где хранятся данные клиентов, но из-за него было

прервано энергоснабжение. В общей сложности пострадало 9 тыс. серверов, на которых хранились данные 7,5 тыс. клиентов провайдера. После пожара данные клиентов разместили в других дата-центрах «The Planet»<sup>[5]</sup>.

## Интересные факты

---

Крупнейший в [СНГ](#) дата-центр будет создан в [Павлодаре \(Казахстан\)](#). Строительство планируется начать в конце [2011 года](#) и завершить в третьем квартале [2012 года](#)<sup>[6]</sup>.

## Размещение оборудования в дата-центрах. Аренда физического и виртуального выделенного сервера.

Вне зависимости от размера бизнеса и направления деятельности предприятие любого масштаба может арендовать или разместить сервер/серверы в дата-центрах компании РТКОММ. На сегодняшний день высокотехнологичные современные дата-центры компании располагаются в четырех городах России. ЦОДы РТКОММ находятся в Москве, Ростове-на-Дону, Новосибирске и Уфе. Размещение оборудования в ЦОД на территории России значительно повышает качество и скорость соединения. Помещение каждого дата-центра оснащено источниками бесперебойного питания, системой кондиционирования, регулирования чистоты и влажности воздуха, телекоммуникационными шкафами, стойками, датчиками контроля за состоянием окружающей среды, системой администрирования, а также дополнительными устройствами физической защиты комнаты.

Для обеспечения высокого уровня безопасности информации РТКОММ использует оборудование ведущих мировых производителей и надежные системы хранения данных: все устройства объединены в RAID-массивы, что исключает потерю информации в случае сбоев в работе сервера.

## Услуги дата-центров РТКОММ

### Colocation (колокейшн)

Colocation — размещение оборудования клиентов на площадях хостинг-провайдера: аренда мест в стойках, аренда стоек или места под них. Возможна как персональная установка нового оборудования, так и перенос сервера из любого ЦОД в дата-центры РТКОММ. Использование услуги колокейшн гарантирует высокую информационную безопасность и качество администрирования.

### Размещение сервера в шкафу (стойке)

В дата-центр (data-center) можно поместить сервер в корпусе стоечного исполнения. Воспользовавшись услугой, клиент получает место для размещения оборудования и гарантированную мощность электропитания в стойках. Месторасположение сервера определяется исходя из высоты корпуса устройства (шаг высоты — [один юнит \(1U\)](#)).

### Выделенный шкаф (аренда стоек)

Большое количество оборудования удобно размещать в отдельном шкафу (стойке). При аренде стойки в дата-центре (data-center) РТКОММ клиент получает в свое распоряжение достаточно места для размещения серверов, единственным ограничением является допустимый уровень потребляемой мощности электропитания (максимум — 5, 6 или 8 кВт). Высота стоек — 42U, шкафов — 47U. В случае необходимости возможен ввод в стойку резервного кабеля электропитания, идущего от основного или дополнительного источника энергоснабжения дата-центра (data-center).

## Colocation оборудования с большим энергопотреблением

При переносе блейд-сервера или дискового массива на площади хостинг-провайдера необходимо заказать специальный шкаф, допустимая потребляемая мощность устройств в котором от 12 до 20 кВт, поскольку технических возможностей обычного телекоммуникационного шкафа для обеспечения стабильной работы оборудования может оказаться недостаточно.

## Аренда выделенного сервера в ЦОД (dedicated хостинг)

В ряде случаев аренда готового к работе выделенного сервера (dedicated server) является более выгодным решением, чем покупка, обслуживание и модернизация собственного серверного оборудования. Компания РТКОММ предлагает клиентам услуги по аренде выделенного сервера с возможностью его последующего выкупа. Воспользовавшись услугой «dedicated server», клиент арендует оборудование в современном высокотехнологичном дата-центре с надежными системами хранения данных, стабильную работу которого обеспечивают специалисты компании РТКОММ.

## Аренда виртуального выделенного сервера (virtual dedicated server, VDS)

Благодаря современным технологиям виртуализации физический сервер легко разделяется на несколько виртуальных выделенных серверов (virtual dedicated server, VDS), каждый из которых имеет собственную операционную систему со своим набором программ и комплексом настроек. Виртуальный выделенный сервер (virtual dedicated server, VDS) обладает всеми основными свойствами физического сервера.

Заказать услугу VDS вы можете в интернет-супермаркете компании РТКОММ [www.oblacom.ru](http://www.oblacom.ru).

## Дополнительные услуги

- Услуги MPLS-сети РТКОММ: выделенные каналы передачи данных, IP-VPN, объединение LAN (AToM Ethernet).
- Регистрация Автономной Системы (AS): оформление заявки в RIPE NCC по присвоению клиенту собственного независимого блока IP-адресов (IPv6) и номера AS.
- Маршрутизация через точки обмена интернет-трафиком (IX).
- Поддержка маршрутизации Автономных Систем клиентов.
- Резервное копирование данных.

К услуге могут быть подключены только серверы, размещенные на площадках РТКОММ. В качестве хранилища данных используются ленточные накопители, расположенные вне дата-центров. Копирование производится внутри сети РТКОММ по выделенному VPN-каналу. Это гарантирует полную конфиденциальность передаваемых данных.

[Виртуальный хостинг \(virtual hosting\)](#) — услуга по размещению сайтов в сети Интернет. [Хостинг](#) - универсальная услуга, которая подойдет как владельцам домашних страничек, так и владельцам интернет-магазинов и корпоративных сайтов.

Если у Вас возникли трудности в выборе хостинга — обратитесь к нам прямо сейчас, и наши специалисты помогут сделать Вам выбор!

Качественный и надежный [Web хостинг](#) — одна из самых важных составляющих успешного и прибыльного интернет-бизнеса. Отсутствие сбоев, высокая скорость загрузки страниц, круглосуточная бесперебойная работа сайта влияет в том числе и на отношение к сайту поисковых систем, а в конечном счете — и на количество посетителей сайта. Все это обеспечивается правильным выбором хостинг-провайдера. Виртуальный хостинг представляет собой размещение сайтов на сервере поставщика услуг web хостинга. При этом на одном и том же сервере может быть размещено большое количество сайтов разных клиентов. Каждому клиенту в соответствии с выбранным тарифом предоставляется определенное количество дискового пространства сервера для хранения собственной информации. Сервера круглосуточно подключены к сети Интернет и размещаются в специально оборудованных помещениях, где поддерживается оптимальная для их работы температура и влажность, а также создана многоступенчатая система защиты от перебоев электрической сети.

Плюсы виртуального хостинга очевидны: это демократичная стоимость размещения сайтов, высокая скорость и надежность соединения, защита от несанкционированного доступа к информации. Однако web хостинг имеет и ряд недостатков, среди которых можно назвать отсутствие возможности установки собственного программного обеспечения на сервер, взаимосвязь всех сайтов, размещенных на сервере, ограниченное дисковое пространство. В случае, если бывает перегружен один из сайтов, на это соответствующим образом реагируют все остальные ресурсы, размещенные на том же [выделенном сервере](#).

Тем не менее, если интернет-проект не предполагает применения сложных ресурсоемких процессов обработки данных или огромного количества посетителей, виртуальный хостинг для размещения сайтов подойдет наилучшим образом. Web хостинг — одно из основных направлений деятельности нашей компании. Мы предоставляем виртуальный хостинг по четырем тарифным планам, различающимся по количеству предоставляемого дискового пространства для размещения сайтов, количеству доменных имен и некоторым другим характеристикам. Наши сервера расположены в специализированном data-центре, где созданы все необходимые условия для их бесперебойной работы. Размещение сайтов на web хостинге позволяет избежать проблем,

связанных с поддержкой и техническим обслуживанием собственного сервера. Виртуальный хостинг – это недорогое и эффективное решение для Вашего бизнеса. Если ваш ресурс вырос, то вам непременно будет интересна услуга [аренда выделенного сервера](#)

Правильный выбор хостинга!

Как и где найти лучший хостинг без проблем? Этот вопрос встаёт сразу после того как мы создали свой первый сайт. Правильный выбор хостинга задача не простая, и очень ответственная.

Здесь как нельзя лучше действует поговорка "семь раз отмерь один раз отрежь". Настоятельно рекомендую перед решающим выбором ознакомиться со всеми предложенными вариантами. И остановиться на том, с кем очень долго и плодотворно будете сотрудничать. Попробуем разобраться, какой же нам выбрать хостинг, и по каким критериям оценивать.

Критерии выбора хостинга.

Первый критерий - аптайм, то есть время непрерывной работы серверов. Важнейший параметр надежности хостинга. Я бы даже сказал критичный, так как от этого параметра зависит Ваше "место под солнцем". Место, которое занимает сайт в выдаче поисковых систем. Но поисковые системы это всё вторично! Ведь мы стремимся понравиться реальным людям! Вот вам нравится когда вы хотите посетить свой любимый ресурс, а он не отвечает? Вот и всем остальным это тоже не нравится. Ещё раз повторюсь время непрерывной работы важнейший параметр оценки качества.

Причем тут аптайм и место в поисковике спросите Вы? Робот ПС пришедший посмотреть, как поживает Ваш ресурс. Будет очень недоволен, если не найдёт его там, где он должен быть, и понизит, а скорее всего совсем выкинет из выдачи до следующего раза, который может быть и через неделю и через месяц.

У всех хостеров, предложенных мною на страницах данного сайта, время непрерывной работы гарантировано составляет 99% и выше.

Второй по значимости особенно для нас, является стоимость, потому как это наш первенец и большого количества денег пока у нас нет. Главное фанатично не увлекаться ценой, ибо можно потерять качество. Нужно найти их оптимальное соотношение.

Ещё один критерий, но не последний, это поддержка нашей CMS(Система управления контентом). Если у нас Joomla или ВордПресс, то мы ищем хостера с установкой данных CMS "одним нажатием". Когда нашу CMS можно установить прямо из панели управления хостера одним кликом! Это очень удобно!

Обратите внимания так же на способы оплаты. На стоимость регистрации доменного имени, и что ещё важнее на его продления через год. Иногда малые цены регистрации, скрывают за собой не малые цены продления домена.

На следующей странице будет подробнейшим образом описан [Самый лучший хостинг в России](#). Вам не интересно? Тогда обязательно посмотрите [рейтинг хостингов](#)!

Вот кратко о правильном выборе хостинга. Определитесь со своими потребностями, что вам в первую очередь важно дешёвизна, надёжность, поддержка вашей CMS(joomla, wordpress) или возраст хостинг компании. Сделайте правильный выбор!

## Правильный выбор хостинга

Выбор хостинга является весьма непростой задачей. И в первую очередь нужно исходить из тех требований, которые предъявляются к сайту. Если ваш сайт имеет огромное количество ежедневных посетителей (например, социальная сеть), то лучше выбирать для его размещения у хостера отдельный выделенный сервер, или, по крайней мере, VPS-хостинг. Для сайтов с низкой и средней посещаемостью вполне можно довольствоваться виртуальным хостингом. К выбору компании, предоставляющей услуги хостинга, надо подходить весьма ответственно. К сожалению, на современном рынке виртуального хостинга сложилась весьма устойчивая тенденция ухудшения качества предоставляемых услуг с течением времени. Объясняется это увеличением количества клиентов хостера без своевременного увеличения парка веб-серверов, в результате чего начинаются простои и зависания серверного оборудования, ведущие к отказу в работе расположенных на нем сайтов. Поэтому, прежде чем оплачивать услуги предполагаемого хостера, почитайте о нем отзывы, например, на сайте [hostobzor.ru](http://hostobzor.ru), или выбирайте хостинг-компанию, предоставляющую возможность бесплатного тестового хостинга. И помните: не всегда дешевый хостинг обозначает плохое его качество, и наоборот, многие компании, безбожно обдирающие своих клиентов, не торопятся предоставлять им соответствующее качество услуг.

И напоследок, вы можете получить жирную ссылку на свой сайт, если разместите отзыв о своем хостинг провайдере на сайте [hostobzor.ru](http://hostobzor.ru). Главное условие – чтобы ваш сайт на текущий момент находился именно у того хостинг-провайдера, о котором вы пишете свой отзыв.

## Как правильно выбрать хостинг

Любой человек желающий поместить свой сайт в интернет, должен правильно выбрать хостинг. Поставщиков хостинга с каждым днём становится всё больше. Соответственно, среди этого многообразия выбрать наилучший, довольно трудно. Далее мы попробуем разобраться, как это сделать.

### Для выбора хостинга нужно

Прежде чем выбрать хостинг, определитесь с какой целью вы хотите поместить свой сайт в сети. От этого зависит выбор между платным и [бесплатным хостингом](#). Следует понимать, что бесплатный хостинг имеет некоторые ограничения. Его владельцу нужно окупить свои затраты на него, поэтому почти все бесплатные хостинги используют размещение рекламы на вашем сайте. В связи с этим для престижных проектов лучше выбирать платный хостинг, цена которого, как правило, не высока и варьируется в разумных пределах. Платный хостинг имеет удобную панель управления, отличается высоким качеством работы технических служб и высокой стабильностью работы в целом.

### Факторы для выбора качественного хостинга

Для того чтобы выбрать действительно качественный хостинг, нужно учесть некоторые особенности:

Выбор тарифов с различным объёмом памяти. Нужно определиться, какой объём памяти необходим вам для размещения своего сайта. При этом следует учитывать его возможное развитие в будущем или количество сайтов, которые вы планируете поместить на данный хостинг.



Количество ресурсов, которые позволяет разместить данный тариф. Этот показатель важен если со временем вы планируете разместить в интернете несколько сайтов. Количество сайтов в большинстве случаев зависит именно от выбранного вами тарифа.

Набор технологий поддерживаемых сервером. У хостера есть определённый набор сервисов, который он может предложить владельцу сайта (например, MySQL, Perl, CGI, PHP). Необходимо выяснить подходят ли конкретно вам эти сервисы. Чтобы определить какие именно технологии вам нужны, необходимо проанализировать тот способ, с помощью которого создавался или будет создаваться ваш сайт.

Предоставляемый трафик. Лучше всего отдать предпочтение безлимитному трафику, ведь ваш ресурс будет раскручиваться и в дальнейшем могут возникнуть определённые трудности. Но не всё так просто. Пользуясь самым дешёвым тарифом, вы не сможете использовать бесчисленное количество места. Как правило, безлимитный трафик также имеет ограничения. Чтобы быть готовым к неожиданностям следует заглянуть в панель управления данным аккаунтом, в котором указан истинный размер трафика.

Оперативная служба поддержки. В идеале служба поддержки должна работать круглые сутки, без выходных и праздников. Хороший support должен помочь вам разобраться с вашей проблемой (если она не требует длительного разбирательства) примерно за час. Слишком длительное ожидание ответа указывает на её неудовлетворительную работу. Служба поддержки обязана уведомлять вас обо всех задержках, связанных с рассмотрением вашего вопроса.

Аптайм или время бесперебойной работы вашего сайта. Очень важный показатель при выборе хостинга, ведь именно от работы сайта и зависит его результативность. Если показатель находится в районе 99%, это хороший знак, если же чуть ниже, стоит хорошо подумать выбирать ли такой хостинг. Некоторые хостеры идут на уловки и указывают значение аптайма в 100%. Это обязательно должно вас насторожить, ведь такого просто не бывает. Проверить реальную цифру аптайма можно самостоятельно с помощью соответствующих сервисов.

Это основные особенности, которые могут помочь вам выбрать качественный хостинг, но если вы рассчитываете на отличный результат, ответственно подойдите к этому делу и тщательно проанализируйте каждого кандидата.

#### Правильный выбор PHP и MySQL хостинга

Задача данного раздела помочь пользователям выбрать хостинг. И не просто выбрать его, а и остаться довольным своим выбором. Наиболее популярной на сегодняшний день услугой, является виртуальный хостинг, поэтому на нем мы остановимся более детально. Российский хостинг можно разделить на следующие сегменты: 37% - персональные страницы пользователей, 45% - сайты малых предприятий, 15% - проекты крупных компаний, 3% - другие веб-проекты. И почти каждый владелец сайта, уже определившись в каком сегменте будет находиться его веб-проект, задается вопросом: как выбрать хостинг для сайта?

Сначала конечно же нужно понять, готовы ли Вы платить за хостинг? Ведь [бесплатный хостинг](#) полезен только для начинающих веб-мастеров, и серьезные задачи ему не под силу. Такой хостинг скорее предназначен для тренировки веб-мастера (загрузка сайта на [сервер](#), [перенос домена](#), работа сайта

на сервере и т.д.). Поэтому если Ваш веб-проект хоть чуточку серьезный, то лучше выбрать хостинг на платной основе. Так Вы получите более лучший хостинг по всем его показателям.

Качественными показателями виртуального хостинга являются:

- поддержка таких технологий как: Perl, PHP, ASP, CGI, htaccess, баз данных (MySQL);
- размер дискового пространства;
- количество сайтов, которые можно разместить;
- ограничение объема посетителей за период времени;
- ограничения по [базам данных MySQL](#) (количеству и объему);
- ресурсы и возможности сервера, которые влияют на его быстродействие.

Важно! Выбирая платный отечественный хостинг, мы настоятельно рекомендуем обращать внимание на то время, которое провайдеры работают на российском рынке хостинга, и не гнаться за низкими ценами. Хотя наши специалисты, составляя рейтинг хостинг компаний, пытаются максимально убедиться в их надежности и состоятельности. Но как показала практика, новым русским компаниям, которые не имеют за своими плечами десятков тысяч постоянных пользователей, очень тяжело "оставаться на плаву". И в этом случае, они рискуют не только своим капиталом, а еще и Вашей информацией, которая хранится на сервере. Ну а окончательный выбор хостинга, конечно остается всегда за Вами, мы лишь можем посоветовать и предоставить максимум информации о крупных хостерах.

#### Такие хостинг-компании лучшие: в сентябре 2012 года

С целью обеспечения будущему веб-сайту плодотворной и долгой жизни весьма важен правильный выбор хостинга. Отнестись к данному процессу нужно серьезно, ведь только лишь качественный хостинг обеспечит сайту бесперебойную и быструю работу.

При выборе хостинга следует обратить внимание на несколько основных критериев:

Выбирая хостинг первый для вашего только что созданного сайта, клиенты первым делом обращают внимание на плату за трафик. Для начала следует выбрать хостинг с бесплатным трафиком с вашего веб-сайта. Такую информацию, о цене или бесплатности трафика, как правило, провайдеры размещают на странице со стоимостью тарифных планов. Чаще всего в цене основного тарифа оплата за трафик не учитывается. В результате чего, вы будете платить больше, чем изначально рассчитывали, однако при условии, что у веб-сайта будет весьма высокая посещаемость.

При выборе хостинга весьма важно учесть так называемый аптайм или время непрерывной работы. Ведь даже наиболее качественные и дорогие сервера не способны работать бесперебойно все время. Всегда нужно учитывать вероятность неожиданной перезагрузки либо же внезапной потери подключения к Интернету. Собственно поэтому существует такой параметр

как аптайм (uptime), показывающий средний за сутки процент нахождения сервера в сети. Чаще всего такой показатель колеблется в пределах 99-99.9%. При выборе хостера, если такого рода информация не указана на сайте провайдера, следует уточнить эти показатели у службы технической поддержки. Не следует заказывать хостинги, у которых показатель аптайма ниже 99%.



Поинтересуйтесь, есть ли возможность организации dedicated серверов и VPS серверов. Ведь, ежели вы планируете создать солидный веб-проект, никогда не помешает имеющаяся возможность организовать собственный виртуальный приватный сервер (VPS) либо же dedicated сервер. Причем для поддержания такого рода сервера стандартного тарифного плана хостинга не будет достаточно.

При выборе хостинга уточните у провайдера наличие собственного дата-центра. Ведь, как правило, многие хостеры арендуют серверы у других провайдеров. Это позволит снизить высокую вероятность внезапного прерывания работы сервера. Однако это не очень существенный недостаток, поскольку множество успешных хостинговых компаний арендуют серверы и оборудование у крупных партнеров.

Также при выборе хостинга стоит обратить внимание на:

автоматическое создание резервных копий баз данных, антивирусной проверки, а также SSH

присутствие оперативной и грамотной службы технической поддержки

наличие в Интернете положительных отзывов

Хостинг магазин Coretek предоставит вашему веб-проекту широкие возможности для дальнейшего развития.

Thursday, July 14, 2011

С целью обеспечения будущему веб-сайту плодотворной и долгой жизни весьма важен правильный выбор хостинга. Отнестись к данному процессу нужно серьезно, ведь только лишь качественный хостинг обеспечит сайту бесперебойную и быструю работу.

При выборе хостинга следует обратить внимание на несколько основных критериев:

Выбирая хостинг первый для вашего только что созданного сайта, клиенты первым делом обращают внимание на плату за трафик. Для начала следует выбрать хостинг с бесплатным

трафиком с вашего веб-сайта. Такую информацию, о цене или бесплатности трафика, как правило, провайдеры размещают на странице со стоимостью тарифных планов. Чаще всего в цене основного тарифа оплата за трафик не учитывается. В результате чего, вы будете платить больше, чем изначально рассчитывали, однако при условии, что у веб-сайта будет весьма высокая посещаемость.

При выборе хостинга весьма важно учесть так называемый аптайм или время непрерывной работы. Ведь даже наиболее качественные и дорогие сервера не способны работать бесперебойно все время. Всегда нужно учитывать вероятность неожиданной перезагрузки либо же внезапной потери подключения к Интернету. Собственно поэтому существует такой параметр как аптайм (uptime), показывающий средний за сутки процент нахождения сервера в сети. Чаще всего такой показатель колеблется в пределах 99-99.9%. При выборе хостера, если такого рода информация не указана на сайте провайдера, следует уточнить эти показатели у службы технической поддержки. Не следует заказывать хостинги, у которых показатель аптайма ниже 99%.



Поинтересуйтесь, есть ли возможность организации dedicated серверов и VPS серверов. Ведь, ежели вы планируете создать солидный веб-проект, никогда не помешает имеющаяся возможность организовать собственный виртуальный приватный сервер (VPS) либо же dedicated сервер. Причем для поддержания такого рода сервера стандартного тарифного плана хостинга не будет достаточно.

При выборе хостинга уточните у провайдера наличие собственного дата-центра. Ведь, как правило, многие хостеры арендуют серверы у других провайдеров. Это позволит снизить высокую вероятность внезапного прерывания работы сервера. Однако это не очень существенный недостаток, поскольку множество успешных хостинговых компаний арендуют серверы и оборудование у крупных партнеров.

Также при выборе хостинга стоит обратить внимание на:

автоматическое создание резервных копий баз данных, антивирусной проверки, а также SSH

присутствие оперативной и грамотной службы технической поддержки

наличие в Интернете положительных отзывов

Хостинг магазин Coretek предоставит вашему веб-проекту широкие возможности для дальнейшего развития.

## Лекция 6. Базы Данных. Системы управления содержимым

### Что такое CMS

Аббревиатура CMS расшифровывается как «Content Management Software» («программное обеспечение для управления содержимым»). В нашей стране принято последнюю букву «S» расшифровывать как «System», а по-русски это обычно звучит как «Система управления контентом». Иногда употребляется более простое название - "движок сайта".

CMS - это специальная программа, которая устанавливается на хостинг-площадке и которая выполняет две основные функции:

**Главная функция CMS** - показывать страницы сайта пользователям, формируя их содержимое "на лету" из заранее определенных шаблонов с дизайном и контента, то есть текстов, картинок, таблиц и других материалов, которые хранятся в базе данных.



Важно понимать, что **сайта как набора страниц при такой схеме просто не существует**. Есть отдельно дизайн (шаблоны) и отдельно набор различных материалов - текст, картинки, файлы с архивами, документы MS Office/PDF и другие материалы. CMS создает страницу пользователю в момент его запроса. При этом, в зависимости от ситуации, пользователю может быть показана какая-то уникальная информация, которая больше никому и никогда не будет видна. Например, содержимое его корзины в интернет-магазине. Эту работу и делает CMS, или "движок сайта". При этом, CMS не просто генерирует страницы пользователям. Она пытается сделать это как можно быстрее, чтобы обработать максимальное количество запросов в единицу времени, защищает данные от недобросовестных пользователей, не дает засорять базу данных спамерам и делает в

фоновом режиме массу дополнительной работы, которая не видна посетителям сайта и даже его владельцу, но жизненно необходима для нормальной работы современного сайта.

Конечно, создать сайт можно и без CMS. Простейшие сайты типа "сайты-визитки" делаются за несколько минут, например с помощью **MS Word**. Для этого достаточно сохранить файл в формат HTML и поместить файл на хостинг-сервер. Правда, когда страниц будет не 1, а 100 возникнут определенные сложности. Если на каждой странице находится, например, телефон компании, то чтобы изменить его потребуются **поменять его в 100 местах**, и все это вручную. Еще хуже, если на страницах нужно будет поменять какую-то ссылку - поддержание такой большой структуры без вспомогательных программных средств будет практически невозможно.

Для того, чтобы добавить на сайт, например, **форму обратной связи**, придется прибегнуть к программированию и **сделать специальный скрипт**, принимающий данные от клиентов и сохраняющий их в базе. Это делает самостоятельное создание сайта для рядового пользователя практически невыполнимой задачей. Но сейчас не 99ый год. Современные CMS позволяют обойтись без необходимости писать самому какой-то программный код и собирать сайт из готовых "кубиков".

**Вторая функция CMS - помочь владельцу сайта без каких-то специальных навыков управлять сайтом**, то есть публиковать новые страницы, новости, выкладывать видео, делать ссылки на внешние ресурсы и так далее. Если войти в админзону, вы увидите примерно вот такой интерфейс:

Чтобы отредактировать любую страницу администратору доступен визуальный редактор, который позволяет форматировать текст, вставлять ссылки и изображения, одновременно видя все в том же виде, как это будет размещено на сайте. Вот как выглядит эта страница в момент написания данной статьи в редакторе:

Как вы можете видеть, редактор сайта не сталкивается ни с какими сложными технологиями при управлении своим сайтом. Он может произвольно менять структуру, добавляя, удаляя и редактируя страницы сайта. Он управляет новостными лентами, каталогами товаров, модерировать форумы и производит другие операции.

#### **Основные функции CMS:**

- Предоставление инструментов для создания контента, организации совместной работы с ним,
- Управление контентом: хранение, **контроль версий**, соблюдение режима доступа, управление потоком документов и т.п.,
- Публикация контента,
- Представление информации в виде, удобном для навигации, поиска.
- В системе управления содержимым могут находиться самые различные данные: документы, фильмы, фотографии, номера телефонов, научные данные и так далее. Такая система часто используется для хранения, управления, пересмотра и публикации документации. **Контроль версий** является одним из основных её преимуществ, когда содержимое изменяется группой лиц.

## Разновидности CMS

В общем случае системы управления содержимым делятся на:

**Систему управления содержанием масштаба предприятия** (Enterprise Content Management System)

**Система управления контентом** (Web Content Management System)

В силу того, что **ECMS** имеют глубокую внутреннюю классификацию по предметным областям, термин CMS заместил собой **WCMS**, превратившись в синоним системы управления сайтами. Подобные CMS позволяют управлять текстовым и графическим наполнением [веб-сайта](#), предоставляя пользователю интерфейс для работы с содержимым сайта, удобные инструменты хранения и публикации информации, автоматизируя процессы размещения информации в базах данных и её выдачи в [HTML](#).

Существует множество готовых систем управления содержимым сайта, в том числе и бесплатных. Их можно разделить на три типа по способу работы:

**Генерация страниц по запросу.** Системы такого типа работают на основе связки «Модуль редактирования → [База данных](#) → Модуль представления». Модуль представления генерирует страницу с содержанием при запросе на него, на основе информации из базы данных. Информация в базе данных изменяется с помощью модуля редактирования. Страницы заново создаются [сервером](#) при каждом запросе, что в свою очередь создаёт дополнительную нагрузку на системные ресурсы.

**Генерация страниц при редактировании.** Системы этого типа суть программы для редактирования страниц, которые при внесении изменений в содержание сайта создают набор статических страниц. При таком способе в жертву приносится [интерактивность](#) между посетителем и содержимым сайта.

**Смешанный тип.** Как понятно из названия, сочетает в себе преимущества первых двух. Может быть реализован путём кэширования — модуль представления генерирует страницу один раз, в дальнейшем она в несколько раз быстрее подгружается из кэша. Кэш может обновляться как автоматически, по истечении некоторого срока времени или при внесении изменений в определённые разделы сайта, так и вручную по команде администратора. Другой подход — сохранение определённых информационных блоков на этапе редактирования сайта и сборки страницы из этих блоков при запросе соответствующей страницы пользователем.

## Особенности

Система управления — программа, предоставляющая инструменты для добавления, редактирования, удаления информации на сайте. Существуют разнообразные системы управления сайтом, среди которых встречаются платные и бесплатные, построенные по разным технологиям. Каждый сайт имеет панель управления, которая является только частью всей программы, но достаточна для управления им.

Термин [контент-менеджер](#) обозначает род профессиональной деятельности — редактор сайта.

Большая часть современных систем управления содержимым реализуется в виде визуального ([WYSIWYG](#)) редактора — программы, которая создаёт [HTML](#)-код из специальной упрощённой разметки, позволяющей пользователю проще форматировать текст.

### Что такое "сборка сайта"?



Для того, чтобы пользователи могли заходить на сайт, а владелец мог им управлять необходимо предварительно создать сам сайт. Условно сайт можно разделить на две части - шаблоны дизайна и контент. Когда вы ходите по сайту, вы видите некоторые элементы, которые не меняются от страницы к странице. Это шапка сайта, подвал, элементы навигации (разные меню, "хлебные крошки" и прочее). Эти блоки обычно помещаются в шаблоны. Остальное - тексты, картинки, таблицы и так далее, как правило, управляется владельцем сайта через CMS. Также, контент может быть произвольным и формируемым по определенным правилам. Например, сейчас вы читаете страницу с произвольным контентом, куда мы, владельцы сайта, можем вставить любую информацию, оформив её так, как нам хочется. Но если вы посмотрите в раздел "Новости", или "Библиотека разработчика", то там информация оформлена определенным образом, как новостная лента и каталог, соответственно. Это функциональные блоки CMS, которые помогают разработчику быстро собирать сайты из готовых кубиков, которые на всех сайтах устроены примерно одинаково, а владельцу сайта не заморачиваться на оформление новостных лент и других блоков, оставив эту работу CMS и только вводя нужную информацию в базу данных.

Процесс верстки шаблонов дизайна и выделения функциональных блоков на сайте называется "сборкой сайта". Как правило, сайт собирается программистами, или веб-мастерами, потому что для этой работы, как правило, требуются особые технические знания. Но зачастую, можно купить или взять бесплатно готовые, собранные заранее сайты-заготовки, в которых уже вся основная работа сделана заранее. Владелец сайта остается только заменить логотип, ввести свой контент и сайт готов.

### Зачем нужны платные CMS, если в интернете есть бесплатные программы?

"Нужно делить системы на производимые **коммерческими компаниями**, и на те, которые делаются **на энтузиазме определенной группы** разработчиков. Для простоты пусть все системы управления сайтом будут или "платные" или "бесплатные".

Защищенность системы никак от платности или бесплатности не зависит. Она зависит от:

- распространенности системы (чем больше пользователей, тем быстрее находятся баги);
- квалификации разработчиков и их желания что-то исправлять;
- возможности команды разработки тестировать продукт - для этого нужно время, тестировщики и соответствующее окружение (набор операционок, браузеров, других настроек для симуляции ситуаций пользователей и воспроизведения ошибок).

**Открытость кода.** Мифы и рассказы про "открытый код, который можно поковырять самому" нужно отнестись с осторожностью. Профессионалов в области безопасности интернет-приложений по пальцам сосчитать, а тех, кто в состоянии грамотно разобрать и собрать чужую программу еще меньше. Если стоит задача сделать её при этом лучше чем была, то это очень долгий и дорогой процесс. Поэтому, открытость и закрытость кода де-факто почти никак не влияют на защищенность системы. Если у системы управления отлажена система поддержки и обновлений, плюс если пользователи этим пользуются (спасибо Microsoft за её идеологию патчей и service packов) - возможные риски эксплуатации любой CMS будут сведены к минимуму.

Вообще, нужно понимать, что если грамотно построить информационную политику по управлению сайтом, отладить и правильно использовать процедуры резервного копирования, разделения доступа, замены паролей и т.п., то риски проникновения на сайт, которые есть в любой системе - коммерческая она или нет - значительно снижаются.

**Абсолютно надежных систем нет.**

### **Чем же реально отличается "платная" CMS от "бесплатной"?**

Производитель платной системы заинтересован в притоке новых клиентов и продолжении сотрудничества со старыми. Поэтому вероятность, что Вашей проблемой будут заниматься конкретные люди в службе поддержки намного выше, чем вероятность, что в форуме бесплатной системы вы найдете ответ на свой неординарный вопрос.

Компании-разработчики платных систем делают их с определенной целью и по определенному плану развития (roadmap). Я оговорюсь - как правило делают. Поэтому в них меньше разных ненужных функций, которые иногда появляются в системе из-за желания разработчика опробовать новую идею или технологию.

**Коммерческие системы,** как правило, разработаны командой, куда помимо программиста входят архитектор, тестировщик, юзабилити проектировщик (его роль могут исполнять проджект менеджер, документатор или системный архитектор) и документатор.

В общем случае это приводит к тому, что системой коммерческого плана:

- удобнее пользоваться;
- она более надежна в работе;
- она более оптимизирована по коду, поскольку в меньшей степени зависит от конкретного разработчика, которые меняются за время разработки системы и не один раз;
- лучше документирована.

Если совсем кратко, то коммерческая система больше сфокусирована на нуждах пользователя, чем бесплатная. Потому, что это программный продукт, который производится для получения денег, а не для получения удовольствия разработчика, удовлетворения его эго и тщеславия.

Вообще, хочется еще отметить, что в настоящее время коммерческие системы только-только догоняют до той границы, где они начинают обладать бесспорными преимуществами перед

бесплатными моделями. Причин тут несколько, но основная - это узость рынка и малый его объем. Но судя по всему, в ближайшее время бесплатники начнут значительно отставать, поскольку новые требования к простоте и удобству использования уже не решить на уровне двухнедельного знания PHP, ASP, Perl или Java. Нужно иметь большой опыт разработок сложных распределенных систем и уйму времени для тестирования и отладки (или большую команду). В чем коммерческие компании, конечно, будут выигрывать."

### Система управления версиями

Система управления версиями (Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы. Однако они могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов. В частности, системы управления версиями применяются в САПР, обычно в составе систем управления данными об изделии. Управление версиями используется в инструментах конфигурационного управления (Software Configuration Management Tools).

### Общие сведения

Ситуация, в которой электронный документ за время своего существования претерпевает ряд изменений, достаточно типична. При этом часто бывает важно иметь не только последнюю версию, но и несколько предыдущих. В простейшем случае можно просто **хранить несколько вариантов документа**, нумеруя их соответствующим образом. Такой **способ неэффективен** (приходится хранить несколько практически идентичных копий), требует повышенного внимания и дисциплины и часто ведёт к ошибкам, поэтому были разработаны средства для автоматизации этой работы.

Традиционные системы управления версиями **используют централизованную модель**, когда имеется единое хранилище документов, управляемое специальным сервером, который и выполняет большую часть функций по управлению версиями. Пользователь, работающий с документами, должен сначала получить нужную ему версию документа из хранилища; обычно создаётся локальная копия документа, т.н. «рабочая копия». Может быть получена последняя версия или любая из предыдущих, которая может быть выбрана по номеру версии или дате создания, иногда и по другим признакам. После того, как в документ внесены нужные изменения, новая версия помещается в хранилище. В отличие от простого сохранения файла, предыдущая версия не стирается, а тоже остаётся в хранилище и может быть оттуда получена в любое время. Сервер может использовать т.н. [дельта-компрессию](#) — такой способ хранения документов, при котором сохраняются только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Поскольку обычно наиболее востребованной является последняя версия файла, система может при сохранении новой версии сохранять её целиком, заменяя в хранилище последнюю ранее сохранённую версию на разницу между этой и последней версией. Некоторые системы (например, ClearCase) поддерживают сохранение версий обоих видов: большинство версий сохраняется в виде дельт, но периодически (по специальной команде

администратора) выполняется сохранение версий всех файлов в полном виде; такой подход обеспечивает максимально полное восстановление истории в случае повреждения **репозитория** (место, где хранятся и поддерживаются какие-либо данные).

Иногда создание новой версии **выполняется незаметно** для пользователя (прозрачно), либо прикладной программой, имеющей встроенную поддержку такой функции, либо за счёт использования специальной файловой системы. В этом случае пользователь просто работает с файлом, как обычно, и при сохранении файла автоматически создаётся новая версия.

Часто бывает, что над одним проектом **одновременно работают несколько человек**. Если два человека изменяют один и тот же файл, то один из них может случайно отменить изменения, сделанные другим. Системы управления версиями отслеживают такие конфликты и предлагают средства их решения. Большинство систем может автоматически объединить (слить) изменения, сделанные разными разработчиками. Однако такое автоматическое объединение изменений, обычно, возможно только для текстовых файлов и при условии, что изменялись разные (непересекающиеся) части этого файла. Такое ограничение связано с тем, что большинство систем управления версиями ориентированы на поддержку процесса разработки программного обеспечения, а исходные коды программ хранятся в текстовых файлах. **Если автоматическое объединение выполнить не удалось, система может предложить решить проблему вручную.**

Часто выполнить слияние невозможно ни в автоматическом, ни в ручном режиме, например, если [формат файла](#) неизвестен или слишком сложен. Некоторые системы управления версиями дают возможность заблокировать файл в хранилище. Блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла (например, средствами файловой системы) и обеспечивает, таким образом, исключительный доступ только тому пользователю, который работает с документом.

Многие системы управления версиями предоставляют ряд других возможностей:

- Позволяют создавать разные варианты одного документа, т.н. ветки, с общей историей изменений до точки ветвления и с разными — после неё.
- Дают возможность узнать, кто и когда добавил или изменил конкретный набор строк в файле.
- Ведут журнал изменений, в который пользователи могут записывать пояснения о том, что и почему они изменили в данной версии.
- Контролируют права доступа пользователей, разрешая или запрещая чтение или изменение данных, в зависимости от того, кто запрашивает это действие.

### **Типичный порядок работы с системой**

Каждая система управления версиями имеет свои специфические особенности в наборе команд, порядке работы пользователей и администрировании. Тем не менее, общий порядок работы для большинства VCS совершенно стереотипен. Здесь предполагается, что проект, каким бы он ни был, уже существует и на сервере размещён его репозиторий, к которому разработчик получает доступ.

### **Начало работы с проектом**

Первым действием, которое должен выполнить разработчик, является извлечение рабочей копии проекта или той его части, с которой предстоит работать. Это действие выполняется с помощью стандартной команды извлечения версии (checkout или clone) либо специальной команды, фактически выполняющей то же самое действие. Разработчик задаёт версию, которая должна быть скопирована, по умолчанию обычно копируется последняя (или выбранная администратором в качестве основной) версия.

По команде извлечения устанавливается соединение с сервером и проект (или его часть — один из каталогов с подкаталогами) в виде дерева каталогов и файлов копируется на компьютер разработчика. Обычной практикой является дублирование рабочей копии: помимо основного каталога с проектом на локальный диск (либо в отдельный, специально выбранный каталог, либо в системные подкаталоги основного дерева проекта) дополнительно записывается ещё одна его копия. Работая с проектом, разработчик изменяет только файлы основной рабочей копии. Вторая локальная копия хранится в качестве эталона, позволяя в любой момент без обращения к серверу определить, какие изменения внесены в конкретный файл или проект в целом и от какой версии была «отпочкована» рабочая копия; как правило, любая попытка ручного изменения этой копии приводит к ошибкам в работе программного обеспечения VCS.

### **Ежедневный цикл работы**

При некоторых вариациях, определяемых особенностями системы и деталями принятого бизнес-процесса, обычный цикл работы разработчика в течение рабочего дня выглядит следующим образом.

**Обновление рабочей копии.** По мере внесения изменений в проект рабочая копия на компьютере разработчика стареет, расхождение её с основной версией проекта увеличивается. Это повышает риск возникновения конфликтных изменений. Поэтому удобно поддерживать рабочую копию в состоянии, максимально близком к текущей основной версии, для чего разработчик выполняет операцию обновления рабочей копии (update) насколько возможно часто (реальная частота обновлений определяется частотой внесения изменений, зависящей от активности разработки и числа разработчиков, а также временем, затрачиваемым на каждое обновление — если оно велико, разработчик вынужден ограничивать частоту обновлений, чтобы не терять время).

**Модификация проекта.** Разработчик модифицирует проект, изменяя входящие в него файлы в рабочей копии в соответствии с проектным заданием. Эта работа производится локально и не требует обращений к серверу VCS.

**Фиксация изменений.** Завершив очередной этап работы над заданием, разработчик фиксирует (commit) свои изменения, передавая их на сервер (либо в основную ветвь, если работа над заданием полностью завершена, либо в отдельную ветвь разработки данного задания). VCS может требовать от разработчика перед фиксацией обязательно выполнить обновление рабочей копии. При наличии в системе поддержки отложенных изменений (shelving) изменения могут быть переданы на сервер без фиксации. Если утверждённая политика работы в VCS это позволяет, то фиксация изменений может проводиться не ежедневно, а только по завершении работы над заданием; в этом случае до завершения работы все связанные с заданием изменения сохраняются только в локальной рабочей копии разработчика.

### **Ветвления**

**Делать мелкие исправления** в проекте можно путём непосредственной правки рабочей копии и последующей фиксацией изменений прямо в главной ветви (стволе) на сервере. Однако при выполнении сколько-нибудь **значительных по объёму работ** такой порядок становится неудобным: отсутствие фиксации промежуточных изменений на сервере не позволяет работать над чем-либо в групповом режиме, кроме того, повышается риск потери изменений при локальных авариях и теряется возможность анализа и возврата к предыдущим вариантам кода в пределах данной работы. Поэтому для таких изменений обычной практикой является создание ветвей (branch), то есть «отпочковывания» от ствола в какой-то версии нового варианта проекта или его части, разработка в котором ведётся параллельно с изменениями в основной версии. Ветвь создаётся специальной командой. Рабочая копия ветви может быть создана заново обычным образом (командой извлечения рабочей копии, с указанием адреса или идентификатора ветви), либо путём переключения имеющейся рабочей копии на заданную ветвь.

Базовый рабочий цикл при использовании ветвей остаётся точно таким же, как и в общем случае: разработчик периодически обновляет рабочую копию (если с ветвью работает более одного человека) и фиксирует в ней свою ежедневную работу. Иногда ветвь разработки так и остаётся самостоятельной (когда изменения порождают новый вариант проекта, который далее развивается отдельно от основного), но чаще всего, когда работа, для которой создана ветвь, выполнена, ветвь реинтегрируется в ствол (основную ветвь). Это может делаться командой слияния (обычно merge), либо путём создания патча (patch), содержащего внесённые в ходе разработки ветви изменения и применения этого патча к текущей основной версии проекта.

### Слияние версий

Три вида операций, выполняемых в системе управления версиями, могут приводить к необходимости объединения изменений:

- Обновление рабочей копии (изменения, сделанные в основной версии, сливаются с локальными).
- Фиксация изменений (локальные изменения сливаются с изменениями, уже зафиксированными в основной версии).
- Слияние ветвей (изменения, сделанные в одной ветви разработки, сливаются с изменениями, сделанными в другой).

Во всех случаях ситуация принципиально одинакова и имеет следующие характерные черты:

Ранее была сделана копия дерева файлов и каталогов репозитория или его части.

Впоследствии и в оригинальное дерево и в копию были независимо внесены некоторые изменения.

Требуется объединить изменения в оригинале и копии таким образом, чтобы не нарушить логическую связность проекта и не потерять данные.

Совершенно очевидно, что при невыполнении условия (2) (то есть если изменения были внесены только в оригинал или только в копию) объединение элементарно — достаточно скопировать изменённую часть туда, где изменений не было. В противном случае слияние изменений превращается в нетривиальную задачу, во многих случаях требующую вмешательства

разработчика. В целом механизм автоматического слияния изменений работает, основываясь на следующих принципах:

Изменения могут состоять в модификации содержимого файла, создании нового файла или каталога, удалении или переименовании ранее существовавшего файла или каталога в проекте.

Если два изменения относятся к разным и не связанным между собой файлам и/или каталогам, они всегда могут быть объединены автоматически. Их объединение состоит в том, что изменения, сделанные в каждой версии проекта, копируются в объединяемую версию.

Создание, удаление и переименование файлов в каталогах проекта могут быть объединены автоматически, если только они не конфликтуют между собой. В этом случае изменения, сделанные в каждой версии проекта, копируются в объединяемую версию. Конфликтующими обычно являются:

- Удаление и изменение одного и того же файла или каталога.
- Удаление и переименование одного и того же файла или каталога (в случае, если система поддерживает операцию переименования).
- Создание в разных версиях файла с одним и тем же именем и разным содержимым.
- Изменения в пределах одного текстового файла, сделанные в разных версиях, могут быть объединены, если они находятся в разных местах этого файла и не пересекаются. В этом случае в объединённую версию вносятся все сделанные изменения.
- Изменения в пределах одного файла, если он не является текстовым, всегда являются конфликтующими и не могут быть объединены автоматически.

Во всех случаях базовой версией для слияния является версия, в которой было произведено разделение сливаемых версий. Если это операция фиксации изменений, то базовой версией будет версия последнего обновления перед фиксацией, если обновление — то версия предыдущего обновления, если слияние ветвей — то версия, в которой была создана соответствующая ветвь. Соответственно, сопоставляемыми наборами изменений будут наборы изменений, сделанных от базовой до текущей версии во всех объединяемых вариантах.

Абсолютное большинство современных систем управления версиями ориентировано, в первую очередь, на проекты разработки программного обеспечения, в которых основным видом содержимого файла является текст. Соответственно, механизмы автоматического слияния изменений ориентируются на обработку текстовых файлов, то есть файлов, содержащих текст, состоящий из строк буквенно-цифровых символов, пробелов и табуляций, разделённых символами перевода строки.

При определении допустимости слияния изменений в пределах одного и того же текстового файла работает типовой механизм построчного сравнения текстов (примером его реализации является системная утилита GNU diff), который сравнивает объединяемые версии с базовой и строит список изменений, то есть добавленных, удалённых и заменённых наборов строк. Минимальной единицей данных для этого алгоритма является строка, даже самое малое отличие делает строки различными. С учётом того, что символы-разделители, в большинстве случаев, не

несут смысловой нагрузки, механизм слияния может игнорировать эти символы при сравнении строк.

Те найденные **наборы изменённых строк, которые не пересекаются** между собой, считаются совместимыми и их слияние делается автоматически. Если в сливаемых файлах находятся изменения, **затрагивающие одну и ту же строку файла**, это приводит к конфликту. Такие файлы могут быть объединены только вручную. Любые файлы, кроме текстовых, с точки зрения VCS являются бинарными и не допускают автоматического слияния.

## Конфликты и их разрешение

Ситуация, когда при слиянии нескольких версий сделанные в них изменения пересекаются между собой, называют конфликтом. При конфликте изменений система управления версиями не может автоматически создать объединённый проект и вынуждена обращаться к разработчику. Как уже говорилось выше, конфликты могут возникать на этапах фиксации изменений, обновления или слияния ветвей. Во всех случаях при обнаружении конфликта соответствующая операция прекращается до его разрешения. Некоторые системы (например, Subversion) даже не пытаются разрешать конфликты на этапе фиксации или слияния ветви со стволом; при возникновении таких конфликтов операция полностью отменяется и разработчику предлагается сначала выполнить обновление рабочей копии (очевидно, что при этом возникнут те же самые конфликты), разрешить конфликты и только после этого выполнять объединение своих изменений с базовой ветвью.

Для разрешения конфликта система, в общем случае, предлагает разработчику три варианта конфликтующих файлов: базовый, локальный и серверный. Конфликтующие изменения либо показываются разработчику в специальном программном модуле объединения изменений (в этом случае там демонстрируются сливаемые варианты и динамически изменяющийся в зависимости от команд пользователя объединённый вариант файла), либо просто помечаются специальной разметкой прямо в тексте объединённого файла (тогда разработчик должен сам сформировать желаемый текст в спорных местах и сохранить его).

Конфликты в файловой системе разрешаются проще: там может конфликтовать только удаление файла с одной из прочих операций, а порядок файлов в каталоге не имеет значения, так что разработчику остаётся лишь выбрать, какую операцию нужно сохранить в сливаемой версии.

## Блокировки

Механизм блокировки позволяет одному из разработчиков захватить в монопольное использование файл или группу файлов для внесения в них изменений. На то время, пока файл заблокирован, он остаётся доступным всем остальным разработчикам только на чтение, и любая попытка внести в него изменения отвергается сервером. Технически блокировка может быть организована по-разному. Типичным для современных систем является следующий механизм.

Файлы, для работы с которыми требуется блокировка, помечаются специальным флагом «блокируемый». Такая пометка может ставиться автоматически при добавлении файла в проект, обычно для этого предварительно создаётся список масок имён файлов, которые при добавлении должны становиться блокируемыми.

Если файл помечен как блокируемый, то при извлечении рабочей копии с сервера он получает в локальной файловой системе атрибут «только для чтения», что препятствует его случайному редактированию.

Разработчик, желающий изменить блокируемый файл, вызывает специальную команду блокировки (lock) с указанием имени этого файла. В результате работы этой команды происходит следующее:

сервер проверяет, не заблокирован ли уже файл другим разработчиком; если это так, то команда блокировки завершается с ошибкой «файл заблокирован другим пользователем» и разработчик, вызывавший её, должен ожидать, пока другой пользователь не снимет свою блокировку;

файл на сервере помечается как «заблокированный», с сохранением идентификатора заблокировавшего его разработчика и времени блокировки;

если блокировка на сервере прошла удачно, на локальной файловой системе с файла рабочей копии снимается атрибут «только для чтения», что позволяет начать его редактировать.

Разработчик работает с заблокированным файлом. Если в процессе работы выясняется, что файл изменять не нужно, он может вызвать команду снятия блокировки (unlock, release lock). Все изменения файла будут отменены, локальный файл вернётся в состояние «только для чтения», с файла на сервере будет снят атрибут «заблокирован» и другие разработчики получат возможность изменять этот файл.

По завершении работы с блокируемым файлом разработчик фиксирует изменения. Обычно блокировка при этом снимается автоматически, хотя в некоторых системах блокировку требуется снимать вручную после фиксации, либо указывать в команде фиксации изменений соответствующий параметр. Так или иначе, при этом файл после изменений теряет флаг «заблокирован» и может быть изменён другими разработчиками.

Массовое использование блокировок, когда все или большинство файлов в проекте являются блокируемыми и для любых изменений необходимо заблокировать соответствующий набор файлов, называется ещё **стратегией «блокированного извлечения»**. Ранние системы управления версиями поддерживали исключительно эту стратегию, предотвращая таким способом появление конфликтов на корню. В современных VCS предпочтительным является использование неблокирующих извлечений, блокировки же считаются скорее неизбежным злом, которое нужно по возможности ограничивать. Недостатки использования блокировок очевидны:

Блокировки просто мешают продуктивной работе, поскольку вынуждают ожидать освобождения заблокированных файлов, хотя в большинстве случаев даже совместные изменения одних и тех же файлов, которые делаются в ходе разных по смыслу работ, не пересекаются и объединяются при слиянии автоматически.

Частота возникновения конфликтов и сложность их разрешения в большинстве случаев не настолько велики, чтобы создать серьёзные затруднения. Возникновение же серьёзного конфликта изменений чаще всего сигнализирует либо о существенном расхождении во мнениях разных разработчиков относительно дизайна одного и того же фрагмента, либо о неправильной организации работы (когда два или более разработчиков делают одно и то же).

Блокировки создают административные проблемы. Типичный пример: разработчик может забыть снять блокировку с занятых им файлов, уходя в отпуск. Для разрешения подобных проблем приходится применять административные меры, в том числе включать в систему технические средства для сброса неверных блокировок, но и при их наличии на приведение системы в порядок расходуется время.

С другой стороны, в некоторых случаях использование блокировок вполне оправданно. Очевидным примером является организация работы с бинарными файлами, для которых нет инструментальных средств слияния изменений либо такое слияние принципиально невозможно (как, например, для файлов изображений). Если автоматическое слияние невозможно, то при обычном порядке работы любое параллельное изменение подобных файлов будет приводить к конфликту. В данном случае гораздо удобнее сделать такой файл блокируемым, чтобы гарантировать, что любые изменения в него будут вноситься только последовательно.

## Версии проекта, теги

Система управления версиями обеспечивает хранение всех существовавших вариантов файлов и, как следствие, всех вариантов проекта в целом, имевших место с момента начала его разработки. Но само понятие «версии» в разных системах может трактоваться двумя различными способами.

Одни системы поддерживают версионность файлов. Это означает, что любой файл, появляющийся в проекте, получает собственный номер версии (обычно — номер 1, условной «нулевой» версией файла считается пустой файл с тем же именем). При каждой фиксации разработчиком изменений, затрагивающих файл, соответствующая часть фиксируемых изменений применяется к файлу и файл получает новый, обычно следующий по порядку, номер версии. Поскольку фиксации обычно затрагивают только часть файлов в репозитории, номера версий файлов, имеющиеся на один и тот же момент времени, со временем расходятся, и проект в целом (то есть весь набор файлов репозитория), фактически, никакого «номера версии» не имеет, поскольку состоит из множества файлов с различными номерами версий. Подобным образом работает, например, система управления версиями CVS.

Для других систем понятие «версия» относится не к отдельному файлу, а к репозиторию целиком. Вновь созданный пустой репозиторий имеет версию 1 или 0, любая фиксация изменений приводит к увеличению этого номера (то есть даже при изменении одного файла на один байт весь репозиторий считается изменённым и получает новый номер версии). Таким способом трактует номера версий, например, система Subversion. Номера версии отдельного файла здесь, фактически, не существует, условно можно считать таковым текущий номер версии репозитория (то есть считать, что при каждом изменении, внесённом в репозиторий, все его файлы меняют номер версии, даже те, которые не менялись). Иногда, говоря о «версии файла» в таких системах, имеют в виду ту версию репозитория, в которой файл был последний раз (до интересующего нас момента) изменён.

Для практических целей обычно имеет значение не отдельный файл, а весь проект целиком. В системах, поддерживающих версионность отдельных файлов, для идентификации определённой версии проекта можно использовать дату и время — тогда версия проекта будет состоять из тех версий входящих в него файлов, которые имелись в репозитории на указанный момент времени. Если поддерживается версионность репозитория в целом, номером версии проекта может выступать номер версии репозитория. Однако оба варианта не слишком удобны, так как ни дата, ни номер версии репозитория обычно не несут информации о значимых изменениях в проекте, о том, насколько долго и интенсивно над ним работали. Для более удобной пометки версий проекта (или его частей) системы управления версиями поддерживают понятиетегов.

Тег (tag) — это символическая метка, которая может быть связана с определённой версией файла и/или каталога в репозитории. С помощью соответствующей команды всем или части файлов проекта, отвечающим определённым условиям (например, входящим в головную версию главной ветви проекта на определённый момент времени) может быть присвоена заданная метка. Таким образом можно идентифицировать версию проекта (версия «XX.XXX.XXX» — это набор версий файлов репозитория, имеющих тег «XX.XXX.XXX»), зафиксировав таким образом его состояние на некоторый желаемый момент. Как правило, система тегов достаточно гибкая и позволяет пометить одним тегом и не одновременные версии файлов и каталогов. Это позволяет собрать «версию проекта» любым произвольным образом. С точки зрения пользователя системы пометка тегами может выглядеть по-разному. В некоторых системах она изображается именно как пометка (тег можно создать, применить к определённым версиям файлов и каталогов, снять). В

других системах (например, Subversion) тег представляет собой просто отдельный каталог на файловом дереве репозитория, куда из ствола и ветвей проекта с помощью команды копирования делаются копии нужных версий файлов. Так что визуально тег — это просто вынесенная в отдельный каталог копия определённых версий файлов репозитория. По соглашению в дерево каталогов, соответствующее тегу, запрещена фиксация изменений (то есть версия проекта, представляемая тегом, является неизменной).

### **Базовые принципы разработки ПО в VCS**

Порядок использования системы управления версиями в каждом конкретном случае определяется техническими регламентами и правилами, принятыми в конкретной фирме или организации, разрабатывающей проект. Тем не менее, общие принципы правильного использования VCS немногочисленны и едины для любых разработок и систем управления версиями.

Любые рабочие, тестовые или демонстрационные версии проекта собираются только из репозитория системы. «Персональные» сборки, включающие ещё незафиксированные изменения, могут делать только разработчики для целей промежуточного тестирования. Таким образом, гарантируется, что репозиторий содержит всё необходимое для создания рабочей версии проекта.

Текущая версия главной ветви всегда корректна. Не допускается фиксация в главной ветви неполных или не прошедших хотя бы предварительное тестирование изменений. В любой момент сборка проекта, проведённая из текущей версии, должна быть успешной.

Любое значимое изменение должно оформляться как отдельная ветвь. Промежуточные результаты работы разработчика фиксируются в эту ветвь. После завершения работы над изменением ветвь объединяется со стволом. Исключения допускаются только для мелких изменений, работа над которыми ведётся одним разработчиком в течение не более чем одного рабочего дня.

Версии проекта помечаются тегами. Выделенная и помеченная тегом версия более никогда не изменяется.

### **Распределённые системы управления версиями**

Также известны как Distributed Version Control System, DVCS. Такие системы используют распределённую модель вместо традиционной клиент-серверной. Они, в общем случае, не нуждаются в централизованном хранилище: вся история изменения документов хранится на каждом компьютере, в локальном хранилище, и при необходимости отдельные фрагменты истории локального хранилища синхронизируются с аналогичным хранилищем на другом компьютере. В некоторых таких системах локальное хранилище располагается непосредственно в каталогах рабочей копии.

Когда пользователь такой системы выполняет обычные действия, такие как извлечение определённой версии документа, создание новой версии и тому подобное, он работает со своей локальной копией хранилища. По мере внесения изменений, хранилища, принадлежащие разным разработчикам, начинают различаться, и **возникает необходимость в их синхронизации**. Такая синхронизация может осуществляться с помощью обмена [патчами](#) или так называемыми **наборами изменений (change sets)** между пользователями.

Описанная модель логически **близка созданию отдельной ветки** для каждого разработчика в классической системе управления версиями (в некоторых распределённых системах перед началом работы с локальным хранилищем нужно создать новую ветвь). Отличие состоит в том, что до момента синхронизации другие разработчики этой ветви не видят. Пока разработчик изменяет только свою ветвь, его работа не влияет на других участников проекта и наоборот. По завершении обособленной части работы, внесённые в ветви изменения, сливают с основной (общей) ветвью. Как при слиянии ветвей, так и при синхронизации разных хранилищ возможны конфликты версий. На этот случай во всех системах предусмотрены те или иные методы обнаружения и разрешения конфликтов слияния.

С точки зрения пользователя, распределённая система отличается необходимостью создавать локальный репозиторий и наличием в командном языке двух дополнительных команд: команды получения репозитория от удалённого компьютера (pull) и передачи своего репозитория на удалённый компьютер (push). Первая команда выполняет слияние изменений удалённого и локального репозитория с помещением результата в локальный репозиторий; вторая — наоборот, выполняет слияние изменений двух репозитория с помещением результата в удалённый репозиторий. Как правило, команды слияния в распределённых системах позволяют выбрать, какие наборы изменений будут передаваться в другой репозиторий или извлекаться из него, исправлять конфликты слияния непосредственно в ходе операции или после её неудачного завершения, повторять или возобновлять неоконченное слияние. Обычно передача своих изменений в чужой репозиторий (push) завершается удачно только при условии отсутствия конфликтов. Если конфликты возникают, пользователь должен сначала слить версии в своём репозитории (выполнить pull), и лишь затем передавать их другим.

Обычно рекомендуется организовывать работу с системой так, чтобы пользователи всегда или преимущественно выполняли слияние у себя в репозитории. То есть, в отличие от централизованных систем, где пользователи передают свои изменения на центральный сервер, когда считают нужным, в распределённых системах более естественным является порядок, когда слияние версий инициирует тот, кому нужно получить его результат (например, разработчик, управляющий сборочным сервером).

**Основные преимущества** распределённых систем — их **гибкость** и значительно бóльшая (по сравнению с централизованными системами) **автономия отдельного рабочего места**. Каждый компьютер разработчика является, фактически, самостоятельным и полнофункциональным сервером, из таких компьютеров можно построить произвольную по структуре и уровню сложности систему, задав (как техническими, так и административными мерами) желаемый порядок синхронизации. При этом каждый разработчик может вести работу независимо, так, как ему удобно, изменяя и сохраняя промежуточные версии документов, пользуясь всеми возможностями системы (в том числе доступом к истории изменений) даже в отсутствие сетевого соединения с сервером. Связь с сервером или другими разработчиками требуется исключительно для проведения синхронизации, при этом обмен наборами изменений может осуществляться по различным схемам.

**К недостаткам** распределённых систем можно отнести **увеличение требуемого объёма дисковой памяти**: на каждом компьютере приходится хранить полную историю версий, тогда как в централизованной системе на компьютере разработчика обычно хранится лишь рабочая копия, то есть срез репозитория на какой-то момент времени и внесённые изменения. Менее очевидным, но неприятным недостатком является то, что в распределённой системе **практически**

**невозможно реализовать некоторые виды функциональности**, предоставляемые централизованными системами:

- **Блокировку файла или группы файлов** (для хранения признака блокировки нужен общедоступный и постоянно находящийся в онлайн центральный сервер). Это вынуждает применять специальные административные меры, если приходится работать с бинарными файлами, непригодными для автоматического слияния.
- **Слежение за определённым файлом или группой файлов** (изменения файлов происходят на разных серверах, слияния и выделения ветвей происходят локально, об изменениях становится известно только при синхронизации, причём не всем разработчикам, а только тем, кто в данной синхронизации участвует).
- **Единая сквозная нумерация версий системы и/или файлов**, в которой номер версии монотонно возрастает (такая нумерация также требует наличия главного сервера, задающего номера версий для всех остальных). В распределённых системах приходится обходиться локальными обозначениями версий и применять тэги, назначение которых определяется соглашением между разработчиками или корпоративными стандартами фирмы.
- **Локальная работа пользователя** с отдельной, небольшой по объёму выборкой из значительного по размеру и внутренней сложности хранилища на удалённом сервере.

Можно выделить следующие типичные ситуации, в которых использование распределённой системы даёт заметные преимущества:

- Периодическая синхронизация нескольких компьютеров под управлением одного разработчика (рабочего компьютера, домашнего компьютера, ноутбука и так далее). Использование распределённой системы избавляет от необходимости выделять один из компьютеров в качестве сервера, а синхронизация выполняется по необходимости, обычно при «пересадке» разработчика с одного устройства на другое.
- Совместная работа над проектом небольшой территориально распределённой группы разработчиков без выделения общих ресурсов. Как и в предыдущем случае, реализуется схема работы без главного сервера, а актуальность репозитория поддерживается периодическими синхронизациями по схеме «каждый с каждым».
- Крупный распределённый проект, участники которого могут долгое время работать каждый над своей частью, при этом не имеют постоянного подключения к сети. Такой проект может использовать централизованный сервер, с которым синхронизируются копии всех его участников. Возможны и более сложные варианты — например, с созданием групп для работы по отдельным направлениям внутри более крупного проекта. При этом могут быть выделены отдельные «групповые» серверы для синхронизации работы групп, тогда процесс окончательного слияния изменения становится древовидным: сначала отдельные разработчики синхронизируют изменения на групповых серверах, затем обновлённые репозитории групп синхронизируются с главным сервером. Возможна работа и без «групповых» серверов, тогда разработчики одной группы синхронизируют изменения между собой, после чего любой из них (например, руководитель группы) передаёт изменения на центральный сервер.

В традиционной «офисной» разработке проектов, когда группа разработчиков относительно невелика и целиком находится на одной территории, в пределах единой локальной компьютерной сети, с постоянно доступными серверами, централизованная система может оказаться лучшим выбором из-за своей более жёсткой структуры и наличия функциональности, отсутствующей в распределённых системах (например, уже упомянутой блокировки). Возможность фиксировать изменения без их слияния в центральную ветвь в таких условиях легко реализуется путём выделения незавершённых работ в отдельные ветви разработки.

### **Системы управления веб-содержимым**

Система управления веб-содержимым (Web Content Management System или WCMS) — программный комплекс, предоставляющий функции создания, редактирования, контроля и организации веб-страниц. WCMS часто используются для создания [блогов](#), личных страниц и [интернет-магазинов](#) и нацелены на пользователей, мало знакомых с программированием.

### **Предоставляемые возможности**

Использование шаблонов отображения, автоматически применяемых к новому или существующему контенту.

Простота редактирования контента благодаря визуальным редакторам. От пользователя не требуется знание языков программирования и разметки;

Расширяемость за счет установки дополнительных модулей и плагинов;

Визуализация содержимого, благодаря которой пользователь может увидеть создаваемый контент до отправки его на сайт (предварительный просмотр).

Автоматическое обновление и следование веб-стандартам;

Управление документооборотом;

Единая категоризация всех видов содержимого (таксономия).

Управление пользователями с различным уровнем доступа

### **Реализация**

Чаще всего WCMS использует базу данных для хранения своих настроек и основного контента.

Многие WCMS используют кэширование, которое ускоряет их работу. Наилучшим образом подходит для сайтов с большой посещаемостью.

WCMS формирует и отображает страницу на основе шаблонов.

Администраторская панель чаще всего представляет собой веб-интерфейс, но некоторые системы используют толстого-клиента.

Открытые WCMS часто состоят из модулей и аддонов.

Конструкторы сайтов, позволяющие самостоятельно создавать, как дизайн сайта, так и его структуру, модули и наполняемый контент.

### **WordPress**

WordPress — система управления содержимым сайта ([CMS](#)) с [открытым исходным кодом](#), распространяемая под [GNU GPL](#). Написана на [PHP](#), в качестве базы данных использует [MySQL](#). Сфера применения — от [блогов](#) до достаточно сложных новостных ресурсов и интернет-магазинов. Встроенная система «тем» и «[плагинов](#)» вместе с удачной архитектурой позволяет конструировать практически любые проекты. WordPress выпущен под лицензией [GPL версии 2](#).

## Достоинства

На 2007 год WordPress была самой популярной системой для ведения автономных [блогов](#)<sup>[1]</sup>.

Среди достоинств WordPress можно выделить следующие:

возможность публикации с помощью сторонних программ и сервисов;

простота установки, настройки;

поддержка [RSS](#), [Atom](#), [trackback](#), [pingback](#);

подключаемые модули (плагины) с уникальной простой системой их взаимодействия с кодом;

поддержка так называемых «тем», позволяющих легко менять как внешний вид, так и способы вывода данных;

«темы» реализованы как наборы [файлов-шаблонов](#) на [PHP](#), что положительно сказывается на скорости и гибкости;

громадные библиотеки «тем» и «плагинов»;

заложенный потенциал архитектуры позволяет легко реализовывать сложные решения;

наличие [ЧПУ](#) (человеко-понятный [URL](#));

наличие русских переводов.

## История

12 июня 2001 года Michel Valdrighi начал разработку «движка» b2, впоследствии к проекту присоединились Matt Mullenweg и Mike Little.<sup>[2]</sup> Автором WordPress считается Matthew Charles Mullenweg ему принадлежат права на трейдмарк WordPress. В 2002 году Matthew Mullenweg поехал в летний лагерь, и на свою видеокамеру Sony наделал кучу снимков которые он выложил в интернет, используя блогový движок b2.

В январе 2003 года автор b2 (Michel Valdrighi) прекратил разработку.

"Мишель Валдриги (Michel Valdrighi) - координатор и главный разработчик проекта практически исчез!

Мы очень переживали и терялись в догадках, что могло с ним случиться.

Лично я неоднократно писал к нему письма, но ни разу не получил ответ. :(

Незаметно уходило время, и дальнейшая разработка проекта b2 повисла в воздухе.

Несколько человек уже создали ветки в попытке добавить функции."[\[3\]](#)-пишет Франсуа Планк основатель b2evolution второй ветки b2.

Matthew Mullenweg также продолжил свою разработку b2, сменил имя на WordPress значительно обновив движок. В 2003 году крупная компания CNET стала использовать WordPress для своих проектов. Matt встречается с вице-президентом компании и принимает предложение о работе с CNET. В 2005 году Matthew Mullenweg ушел из CNET, основал Automattic и посвятил себя разработке проектов с WordPress.

PHP (PHP: Hypertext Preprocessor — «PHP: [препроцессор гипертекста](#)»); первоначально Personal Home Page Tools [\[3\]](#) — «Инструменты для создания персональных веб-страниц») — [скриптовый язык](#) [\[4\]](#) программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством [хостинг-провайдеров](#) и является одним из лидеров среди языков программирования, применяющихся для создания [динамических веб-сайтов](#). [\[5\]](#)

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с [открытым кодом](#). [\[6\]](#) Проект распространяется под [собственной лицензией](#), несовместимой с [GNU GPL](#).

Область применения

В области [программирования для сети Интернет](#) PHP — один из популярных [скриптовых языков](#) (наряду с [JSP](#), [Perl](#) и языками, используемыми в [ASP.NET](#)) благодаря своей простоте, скорости выполнения, богатой функциональности, [кроссплатформенности](#) и распространению исходных кодов на основе [лицензии PHP](#).

Популярность в области построения [веб-сайтов](#) определяется наличием большого набора встроенных средств для разработки веб-приложений [\[7\]](#). Основные из них:

автоматическое извлечение [POST](#) и [GET](#)-параметров, а также переменных окружения веб-сервера в предопределённые массивы;

взаимодействие с большим количеством различных систем управления базами данных ([MySQL](#), [MySQLi](#), [SQLite](#), [PostgreSQL](#), [Oracle \(OCI8\)](#), [Oracle](#), [Microsoft SQL Server](#), [Sybase](#), [ODBC](#), [mSQL](#), [IBM DB2](#), [Cloudscape](#) и [Apache Derby](#), [Informix](#), [Ovrimos SQL](#), [Lotus Notes](#), [DB++](#), [DBM](#), [dBase](#), [DBX](#), [FrontBase](#), [FilePro](#), [Ingres II](#), [SESAM](#), [Firebird](#) / [InterBase](#), [Paradox File Access](#), [MaxDB](#), [Интерфейс PDO](#));

автоматизированная отправка [HTTP-заголовков](#);

работа с HTTP-авторизацией;

работа с [cookies](#) и сессиями;

работа с локальными и удалёнными файлами, [сокетам](#);

обработка файлов, загружаемых на сервер;

работа с [XForms](#).

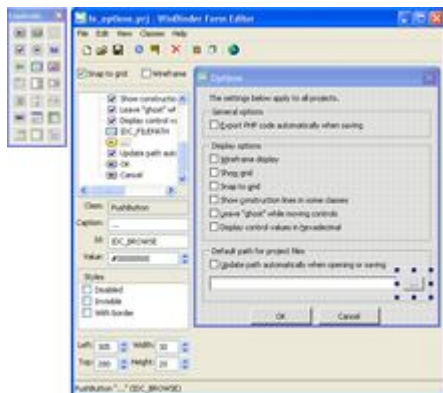
В настоящее время PHP используется сотнями тысяч разработчиков. Согласно рейтингу корпорации TIOBE, базирующемуся на данных поисковых систем, в апреле 2011 года PHP находился на 5 месте среди языков программирования. [\[5\]](#) К крупнейшим сайтам, использующим PHP, относятся [Facebook](#), [ВКонтакте](#), [Wikipedia](#) и др.

Входит в [LAMP](#) — распространённый набор программного обеспечения для создания и [хостинга веб-сайтов](#) ([Linux](#), [Apache](#), [MySQL](#), PHP).

[\[править\]](#)Создание GUI-приложений

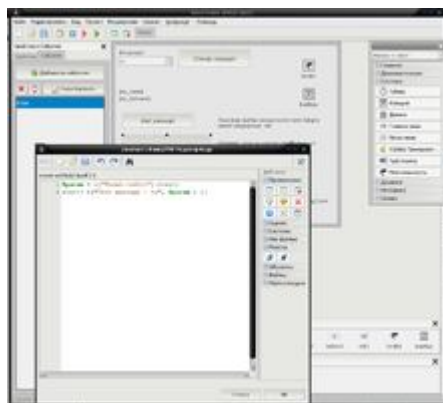
Хотя PHP и не слишком распространён в данном качестве, его можно использовать и для создания [GUI](#)-приложений.

Для создания кроссплатформенных приложений служат пакеты [PHP-GTK](#) и [PHP-Qt](#), представляющие собой обёртки для соответствующих популярных библиотек [виджетов](#).



Скриншот редактора форм WinBinder

Для тех, кого интересует программирование с использованием [Windows API](#), существует две альтернативы. Во-первых, это [open source](#) пакет WinBinder[8]. Его ядро представляет собой написанное на [C](#) расширение php — php\_winbinder.dll. В состав WinBinder включён также визуальный редактор форм (см. скриншот), написанный с использованием самого WinBinder. Но, по сути, WinBinder является простой обёрткой к [WinAPI](#), и программирование с его использованием — достаточно низкоуровневое.



Скриншот среды программирования DevelStudio

Второй альтернативой является [интегрированная среда](#) Devel Studio[9], ориентированная, прежде всего, на начинающих программистов.[10]

Также существует реализация PHP для [.NET/Mono](#) под названием [Phalanger](#). Результатом компиляции PHP кода в Phalanger может быть любое .NET-приложение, будь то серверное или десктопное.

[\[править\]](#)История

Основная статья: [История PHP](#)

В [1994 году датский программист Расмус Лерддорф](#) создал набор [скриптов](#) на [Perl/CGI](#) для вывода и учёта посетителей его [онлайн-резюме](#), обрабатывающий шаблоны [HTML](#)-документов. Лерддорф назвал набор Personal Home Page (Личная Домашняя Страница). Вскоре функциональности и скорости Perl — интерпретатора скриптов — перестало хватать, и Лерддорф разработал с использованием языка [C](#) новый интерпретатор шаблонов PHP/FI ([англ.](#) Personal Home Page / Forms Interpreter — «Личная Домашняя Страница / Интерпретатор форм»).

В [1997 году](#) после длительного [бета-тестирования](#) вышла вторая версия обработчика, написанного на [C](#) — PHP/FI 2.0. Её использовали около 1% (приблизительно 50 тысяч) всех [интернет-доменов](#) мира.[\[3\]](#)

Версия PHP 3.0 подверглась значительной переработке, определившей современный облик и стиль языка программирования. В [1997 году](#) два [израильских программиста](#), [Энди Гутманс](#) и [Зеев Сураски](#), полностью переписали код интерпретатора. PHP 3.0 был официально выпущен в июне [1998 года](#).[\[3\]](#)

Одной из сильнейших сторон PHP 3.0 была возможность расширения ядра дополнительными модулями. Впоследствии интерфейс написания расширений привлёк к PHP множество сторонних разработчиков, работающих над своими модулями, что дало PHP возможность работать с огромным количеством [баз данных](#), [протоколов](#), поддерживать большое число [API](#). Большое количество разработчиков привело к быстрому развитию языка и стремительному росту его популярности. С этой версии акроним php расшифровывается как «PHP: hypertext Preprocessor», вместо устаревшего «Personal Home Page».

К зиме [1998 года](#), практически сразу после официального выхода PHP 3.0, Энди Гутманс и Зеев Сураски начали переработку ядра PHP. В задачи входило увеличение производительности сложных приложений и улучшение модульности базиса кода PHP. Новый движок, названный [Zend Engine](#), успешно справлялся с поставленными задачами и впервые был представлен в середине [1999 года](#). PHP 4.0, основанный на этом движке и принёсший с собой набор дополнительных функций, официально вышел в мае [2000 года](#). В дополнение к улучшению производительности, PHP 4.0 имел ещё несколько ключевых нововведений, таких как поддержка сессий, [буферизация](#) вывода, более безопасные способы обработки вводимой пользователем информации и несколько новых языковых конструкций.

Пятая версия PHP была выпущена разработчиками [13 июля 2004 года](#). Изменения включают обновление ядра Zend (Zend Engine 2), что существенно увеличило эффективность [интерпретатора](#). Введена поддержка языка разметки [XML](#). Полностью переработаны функции [ООП](#), которые стали во многом схожи с моделью, используемой в [Java](#). В частности, введён [деструктор](#), [открытые](#), [закрытые и защищённые члены и методы](#), окончательные члены и методы, [интерфейсы](#) и клонирование объектов. В последующих версиях также были введены [пространства имён](#), [замыкания](#) и целый ряд достаточно серьёзных изменений, количественно и качественно сравнимых с теми, которые появились при переходе на PHP 5.0.

Шестая версия PHP разрабатывалась[\[11\]](#) с [октября 2006](#) года. Было сделано[\[12\]\[13\]](#) множество нововведений, как, например, исключение из ядра регулярных выражений [POSIX](#) и «длинных» суперглобальных массивов, удаление директив `safe_mode`, `magic_quotes_gpc` и `register_globals` из конфигурационного файла `php.ini`. Одним из основных новшеств должна была стать

поддержка [Юникода](#). [14]. Однако в [марте 2010](#) года разработка PHP6 была признана бесперспективной [15] из-за сложностей с поддержкой [Юникода](#). Исходный код PHP6 перемещён на [ветвь](#), а основной линией разработки стала версия 5.4.

[[править](#)]Синтаксис



Имеется [викиучебник](#) по теме  
«[PHP](#)»

Синтаксис PHP подобен синтаксису языка [Си](#). Некоторые элементы, такие как [ассоциативные массивы](#) и цикл foreach, заимствованы из [Perl](#).

Для работы программы не требуется описывать какие-либо переменные, используемые модули и т. п. Любая программа может начинаться непосредственно с оператора PHP.

Простейшая [программа Hello world](#) на PHP выглядит следующим образом:

```
<?php  
  
echo 'Hello, world!';  
  
?>
```

PHP исполняет код, находящийся внутри ограничителей, таких как <?php ?>. Всё, что находится вне ограничителей, выводится без изменений. В основном это используется для вставки PHP-кода в HTML-документ, например, так:

```
<html>  
  
<head>  
  
<title>Тестируем PHP</title>  
  
</head>  
  
<body>  
  
<?php echo 'Hello, world!'; ?>  
  
</body>  
  
</html>
```

Помимо ограничителей <?php ?>, допускается использование дополнительных вариантов, таких как <? ?> и <script language="php"> </script>. Кроме того, до версии 6.0 допускается использование ограничителей [языка программирования ASP](#) <% %> (конструкции <? ?> и <% %> могут быть выключены в конфигурационном файле php.ini).

Имена переменных начинаются с символа \$, тип переменной объявлять не нужно. Имена переменных, функций и классов чувствительны к регистру. Константы также чувствительны к

регистру. Переменные обрабатываются в [строках](#), заключённых в апострофы или двойные кавычки, и [heredoc](#)-строках (строках, созданных при помощи оператора <<<).

PHP рассматривает переход на новую строку как пробел, так же как HTML и другие языки со свободным форматом. Инструкции разделяются с помощью точки с запятой (;), за исключением некоторых случаев, после объявления конструкции if/else и циклов.

PHP поддерживает три типа комментариев: в стиле языка [Си](#) (ограниченные /\* \*/), [C++](#) (начинающиеся с // и идущие до конца строки) и [оболочки UNIX](#) (с # до конца строки).

[\[править\]](#) Типы данных

PHP является языком программирования с [динамической типизацией](#), не требующим указания типа при объявлении переменных, равно как и самого объявления переменных. Преобразования между скалярными типами зачастую осуществляются неявно без дополнительных усилий (впрочем, PHP предоставляет широкие возможности и для явного преобразования типов).

К скалярным типам данных относятся:

[целый тип](#) (integer),

[вещественный тип данных](#) (float, double),

[логический тип](#) (boolean),

[строковый тип](#) (string),

и специальный тип [NULL](#).

К нескаларным типам относятся:

«ресурс» (resource),

[массив](#) (array),

[объект](#) (object),

К псевдотипам [\[16\]](#) относятся:

[mixed](#) один или несколько необязательных параметров,

[number](#) число (integer либо float)

[callback](#) (string или [анонимная функция](#))

[void](#) отсутствие параметров

Диапазон целых чисел (integer) в PHP зависит от [платформы](#) (обычно, это диапазон 32-битных знаковых целых чисел, то есть, от -2 147 483 648 до 2 147 483 647). Числа можно задавать в десятичной, [восьмеричной](#) и [шестнадцатеричной](#) системах счисления. Диапазон вещественных чисел (double), также, зависит от платформы (для 32-битной архитектуры диапазон позволяет оперировать числами от  $\pm 1.7 \times 10^{-308}$  до  $\pm 1.7 \times 10^{308}$ ).

PHP предоставляет разработчикам [логический тип](#) (boolean), способный принимать только два значения TRUE («истина») и FALSE («ложь»). При преобразовании в логический тип число 0, пустая строка, ноль в строке «0», NULL и пустой массив считаются равными FALSE. Все остальные значения автоматически преобразуются в TRUE.

Специальный тип NULL предназначен для переменных без определённого значения. Единственным значением данного типа является константа NULL. Тип NULL принимают неинициализированные переменные, переменные, инициализированные константой NULL, а также переменные, удалённые при помощи конструкции unset().

Ссылки на внешние ресурсы имеют тип «ресурс» (resource). Переменные данного типа, как правило, представляют собой [дескриптор](#), позволяющий управлять внешними объектами, такими как файлы, динамические изображения, результирующие таблицы базы данных и т. п.

[Массивы](#) (array) поддерживают числовые и строковые ключи и являются [гетерогенными](#). Массивы могут содержать значения любых типов, включая другие массивы. Порядок элементов и их ключей сохраняется. Не совсем корректно называть php-массивы массивами, на самом деле это, скорее всего, упорядоченный [хеш](#). Возможно неожиданное поведение при использовании цикла for со счетчиком вместо foreach. Так, например, при сортировке массива с численными индексами функциями из стандартной библиотеки, сортируются и ключи тоже.

Указатель на функцию в PHP может быть представлен замыканием или псевдотипом [callback](#). Замыкание доступно с версии 5.3 и в коде выглядит как простое определение функции, в которую явно можно утянуть значения из контекста, например:

```
function($args...$argsN) use($ctxVar,$ctxVar1) { definition ; }
```

callback тип может быть представлен:

строкой (интерпретируется как название функции);

массивом где нулевой и первый элемент — строки (интерпретируется как название статичной функции в классе);

массивом где нулевой элемент — объект, а первый — строка (интерпретируется как метод у объекта).

Для проверки является ли значение вызываемым следует использовать is\_callable(\$var)

[\[править\]](#)Обращение к переменным и функциям

Обращение к переменным осуществляется с помощью символа \$, за которым следует имя переменной. Данная конструкция может быть применена также для создания динамических переменных и функций. [\[17\]](#) Например:

```
$a = 'I am a';    // Запись значения в переменную $a
```

```
echo $a;         // Вывод переменной $a
```

```
$b = 'a';
```

```
echo $$b;      // Вывод переменной $a (дополнительный $ перед переменной $b)
```

```
echo ${'a'};   // Вывод переменной $a
```

```
function_name(); // Вызов функции function_name
```

```
$c = 'function_name';
```

```
$c();          // Вызов функции function_name,
```

```
$d = 'Class_name';
```

```
$obj = new Class_name; // Создание объекта класса Class_name
```

```
$obj = new $d();       // Создание объекта класса Class_name
```

```
$obj->b; // Обращение к полю b объекта
```

```
$obj->c(); // Вызов метода c() объекта
```

```
$obj->$b; // Обращение к полю a объекта, так как $b = 'a'
```

```
$obj->$c(); // Вызов метода function_name() объекта, так как $c = 'function_name'
```

В PHP echo и print не являются функциями [\[18\]](#) (хотя print имеет возвращаемое значение), а являются синтаксическими единицами. При их использовании можно опустить скобки.

[\[править\]](#)Суперглобальные массивы

Суперглобальными массивами ([англ.](#) Superglobal arrays) в PHP называются predefined массивы, имеющие глобальную область видимости без использования директивы global. Большая часть этих массивов содержит входные данные запроса пользователя (параметры [GET](#)-запроса, поля форм при посылке методом [POST](#), [куки](#) и т. п.).

Все суперглобальные массивы, кроме \$GLOBALS и \$\_REQUEST, имеют устаревшие аналоги с длинными именами, которые доступны вплоть до пятой версии PHP (в шестой версии планируется их исключение). Таким образом, обращения \$\_GET['year'] и \$HTTP\_GET\_VARS['year'] идентичны (за исключением области видимости: массивы с «длинными» именами не являются суперглобальными).

**\$GLOBALS**

Массив всех глобальных переменных (в том числе и пользовательских).

**\$\_SERVER** (устаревший аналог — \$HTTP\_SERVER\_VARS)

Содержит переменные окружения, которые операционная система передаёт серверу.

`$_ENV` (уст. `$HTTP_ENV_VARS`)

Текущие [переменные среды](#) ([англ.](#) Environment variables). Их набор специфичен для платформы, на которой выполняется скрипт.

`$_GET` (уст. `$HTTP_GET_VARS`)

Содержит параметры GET-запроса, переданные в URI после знака вопроса «?».

`$_POST` (уст. `$HTTP_POST_VARS`)

Ассоциативный массив значений полей HTML-формы при отправке методом POST. Индексы элементов соответствуют значению атрибута name элементов управления HTML-формы.

`$_FILES` (уст. `$HTTP_POST_FILES`)

Ассоциативный массив со сведениями об отправленных методом POST файлах. Каждый элемент имеет индекс, идентичный значению атрибута «name» в форме, и, в свою очередь, также является массивом со следующими элементами:

['name'] — исходное имя файла на компьютере пользователя.

['type'] — указанный агентом пользователя [MIME](#)-тип файла. PHP не проверяет его, и поэтому нет никаких гарантий, что указанный тип соответствует действительности.

['size'] — размер файла в байтах.

['tmp\_name'] — полный путь к файлу во временной папке. Файл необходимо переместить оттуда функцией `move_uploaded_file`. Загруженные файлы из временной папки PHP удаляет самостоятельно.

['error'] — код ошибки. Если файл удачно загрузился, то этот элемент будет равен 0 (UPLOAD\_ERR\_OK).

`$_COOKIE` (уст. `$HTTP_COOKIE_VARS`)

Ассоциативный массив с переданными агентом пользователя значениями [куки](#).

`$_REQUEST`

Содержит элементы из массивов `$_GET`, `$_POST`, `$_COOKIE`. С версии PHP 4.1 включает `$_FILES`.

`$_SESSION` (уст. `$HTTP_SESSION_VARS`)

Содержит данные сессии.

[\[править\]](#)Объектно-ориентированное программирование

PHP поддерживает широкие [объектно-ориентированные](#) возможности, полная поддержка которых была введена в пятой версии языка.

[Класс](#) в PHP объявляется с помощью ключевого слова `class`. Методы и поля класса могут быть общедоступными (`public`, по умолчанию), защищёнными (`protected`) и скрытыми (`private`). PHP

поддерживает все три основных механизма ООП —

[инкапсуляцию](#), [полиморфизм](#) и [наследование](#) (родительский класс указывается с помощью ключевого слова `extends` после имени класса). Поддерживаются [интерфейсы](#) (ставятся в соответствие с помощью `implements`). Разрешается объявление [финальных](#), [абстрактных методов](#) и классов. [Множественное наследование](#) классов не поддерживается, однако класс может реализовывать несколько интерфейсов. Для обращения к методам родительского класса используется ключевое слово `parent`.

Начиная с версии 5.4.0 множественное наследование может быть реализовано с помощью механизма особенностей ([англ.](#) `trait`). Особенности похожи на примеси ([англ.](#) `mixins`), за исключением того что для них нельзя напрямую создать экземпляр. Повторное использование кода заключено в использовании кода особенности в нескольких классах. Допускается использовать в одном классе несколько особенностей. Механизм особенностей имеет средства разрешения конфликтов имён. При запуске программы код особенности будет «вкомпилирован» в код содержащего его класса.

Классы в PHP имеют ряд специальных методов ([англ.](#) `magic methods`), начинающихся с двух символов подчёркивания. Особо стоит отметить [конструктор](#) (`__construct()`, в версиях до 5.0 конструктором служил метод, одноимённый с классом) и [деструктор](#) (`__destruct()`), а также методы чтения (`__get()`) и записи (`__set()`), свёртывания (`__sleep()`) и развёртывания (`__wake()`), клонирования (`__clone()`) и др. Эти методы являются достаточно гибким инструментом: переопределяя их, можно добиться существенного изменения поведения объекта.

Экземпляры класса создаются с помощью ключевого слова `new`, обращение к полям и методам объекта производится с использованием оператора `->`. Для доступа к членам класса из его методов используется переменная `$this`.

```
class C1 extends C2 implements I1, I2
```

```
{
```

```
    private $a;
```

```
    protected $b;
```

```
    function __construct($a, $b)
```

```
    {
```

```
        parent::__construct($a, $b);
```

```
        $this->a = $a;
```

```
        $this->b = $b;
```

```
    }
```

```
    public function plus()
```

```

{
    return $this->a + $this->b;
}

/* ..... */
}

```

```

$d = new C1(1, 2);

echo $d->plus(); // 3

```

Начиная с пятой версии PHP, объекты передаются по ссылке:

```

class a
{
    public $color = 'red';
}

```

```

$a = new a();

echo $a -> color; // red

$b = $a;

$b -> color = 'blue';

echo $a -> color; // blue

```

«Paamayim Nekudotayim» ([иврит: פאמ״ק נקודות](#)) или просто «двойное двоеточие». Используя эту [лексемию](#), программист может обращаться к константам, статическим или перегруженным свойствам или методам класса. При обращении к этим элементам извне класса, программист должен использовать имя этого класса. «Paamayim Nekudotayim» на первый взгляд может показаться странным словосочетанием для обозначения двойного двоеточия. Однако, во время создания Zend Engine версии 0.5 (который входил в PHP3), Andi и Zeev выбрали [\[19\]](#) именно это обозначение. «Paamayim Nekudotayim» действительно значит «двойное двоеточие» на иврите. Это обозначение не менялось ни разу в течение всего времени разработки PHP. [\[20\]](#)

```

<?php

class MyClass {

    const CONST_VALUE = 'Значение константы';

}

```

```
// Использование :: вне объявления класса
```

```
echo MyClass::CONST_VALUE;
```

```
?>
```

[[правиль](#)] Особенности интерпретатора

PHP-скрипты обычно обрабатываются [интерпретатором](#) в порядке [\[1\]](#), обеспечивающем [кроссплатформенность](#) разработанного приложения:

[лексический анализ исходного кода](#) и генерация [лексем](#),

[синтаксический анализ](#) полученных лексем,

генерация [байт-кода](#),

выполнение байт-кода интерпретатором (без создания [исполняемого файла](#)).

Для увеличения быстродействия приложений возможно использование специального [программного обеспечения](#), так называемых [акселераторов](#). Принцип их работы заключается в [кэшировании](#) однажды сгенерированного байт-кода в [памяти](#) и/или на [диске](#), таким образом, из процесса работы приложения исключаются этапы 1—3, что в общем случае ведёт к значительному ускорению работы.

Важной особенностью является то, что разработчику нет необходимости заботиться о распределении и [освобождении](#) памяти. Ядро PHP реализует средства для автоматического управления памятью; вся выделенная память возвращается системе после завершения работы скрипта. [\[21\]](#)

[[правиль](#)] Расширения

Интерпретатор состоит из [ядра](#) и подключаемых [модулей](#), «расширений», представляющих собой динамические [библиотеки](#). Расширения позволяют дополнить базовые возможности языка, предоставляя возможности для работы с [базами данных](#), [сокетам](#), динамической [графикой](#), [криптографическими](#) библиотеками, документами формата [PDF](#) и тому подобным. Любой желающий может разработать своё собственное расширение и подключить его. Существует огромное количество расширений, как стандартных, так и созданных сторонними компаниями и энтузиастами, однако в стандартную поставку входит лишь несколько десятков хорошо зарекомендовавших себя. Множество расширений доступно в репозитории [PECL](#).

[[правиль](#)] Параметры настройки

Интерпретатор PHP имеет специальный [конфигурационный файл](#) — `php.ini`, содержащий множество настроек, изменение которых влияет на поведение интерпретатора. [\[22\]](#) Имеется возможность отключить использование ряда функций, изменить ограничения на используемую скриптом оперативную память, время выполнения, объём загружаемых файлов, настроить журналирование ошибок, работу с сессиями и почтовыми сервисами, подключить дополнительные расширения, а также многое другое. Во время выполнения скрипта эти параметры могут быть изменены командой `ini_set`. [\[23\]](#)

[\[править\]](#)Режимы интерпретатора

Существует несколько способов использования интерпретатора PHP:

В качестве модуля к [веб-серверу](#) посредством [SAPI](#) или [ISAPI](#)<sup>[24]</sup> (например, для [Apache](#) модуль mod\_php). Веб-сервер запускается с загруженным модулем и при обращении к php-скрипту он выполняется в [окружении процесса](#) веб-сервера. Это наиболее распространённый метод использования PHP в силу более простой конфигурации.

В качестве [CGI](#). В этом случае, при вызове скрипта веб-сервер вызывает /usr/bin/php-cgi /path/to/script.php. При этом создаётся новый процесс и скрипт выполняется в окружении php-cgi, которое, в зависимости от настроек, может сильно отличаться от окружения процесса веб-сервера. Более современными вариантами являются [FastCGI](#) и [SCGI](#). В отличие от постоянного вызова интерпретатора как в CGI, у FastCGI/SCGI интерпретатор PHP перезапускается не при каждом запросе, а только если все текущие FastCGI-процессы уже в работе, которые освободившись также не завершатся, а будут ждать новых задач. Однако эти процессы опрашиваются один раз в определенный Scan-интервал, согласно настройкам. Процессы, превысившие время жизни, время простоя или отработавшие заданное количество запросов, будут завершаться принудительно, если их количество не ниже минимально указанного для класса (виртхоста).

В качестве [скрипта командной строки](#), являющегося [исполняемым файлом](#), который вызывается пользователем из [командной строки](#); скрипт выполняется в окружении вызвавшего пользователя. В этом случае возможно использование PHP для создания клиентских [GUI-приложений](#)<sup>[25]</sup> и решения административных задач в [операционных системах UNIX](#), [Linux](#), [Microsoft Windows](#), [Mac OS X](#) и [AmigaOS](#). Однако, в таком качестве он не получил распространение, отдавая пальму первенства [Perl](#), [Python](#) и [VBScript](#)

[\[править\]](#)Пасхальные яйца

PHP имеет несколько [пасхальных яиц](#) (шуток). Передача специальных GET-параметров любому php-скрипту приводит к выводу различных изображений.

Например:

любой\_сценарий.php?=PHPE9568F36-D428-11d2-A769-00AA001ACF42

введя это значение в адресную строку браузера можно увидеть различные изображения, в зависимости от версии интерпретатора:

фотография Thies Arntzen (версии 4.1.2, 4.2.2)

фотография [таксы](#) (владелец Stig Bakken) (версии 4.3.2, 4.3.3, 4.3.8 — 4.3.10)

фотография чёрного [терьера](#) (владелец Zeev Suraski) (версии 4.3.11, 4.4.0 — 4.4.4, 5.1.0, 5.1.2)

фотография [кролика](#) (версии 4.3.1, 5.0.0, 5.0.3)

искажённый логотип PHP (версии 5.1.3 — 5.2.13)<sup>[26]</sup>

фотография игрушечного синего слона с надписью PHP (начиная с версии 5.3.0)<sup>[26]</sup>

Следующие два значения позволят увидеть стандартный логотип PHP и ZEND, соответственно:[26]

любой\_сценарий.php?=PHPE9568F34-D428-11d2-A769-00AA001ACF42

любой\_сценарий.php?=PHPE9568F35-D428-11d2-A769-00AA001ACF42

Эти значения можно получить и непосредственно в PHP-скрипте с помощью функций `php_logo_guid` и `zend_logo_guid` для логотипа Zend.

Также имеется специальное значение для отображения списка всех авторов PHP (4,5):[26]

любой\_сценарий.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000

[править]Альтернативные реализации

В силу популярности языка PHP и желания увеличить быстродействие основанных на нём [веб-приложений](#), создано несколько альтернативных [компиляторов](#), близких к PHP языку. Так в феврале [2010 года](#) компания [Facebook](#) открыла[27][28] свой компилятор PHP — [HipHop](#) (HPHP, Hyper-PHP) генерирующий код на [C++](#), с последующей компиляцией в машинный код с помощью [gcc](#).

В таблице представлен список существующих на сегодняшний момент альтернативных реализаций.

| Название                                                               | Лицензия                                                   | Результат компиляции                               |
|------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------|
| <a href="#">HipHop</a>                                                 | <a href="#">PHP License</a>                                | <a href="#">C++</a> , <a href="#">машинный код</a> |
| <a href="#">Roadsend PHP</a>                                           | <a href="#">GPL/LGPL</a>                                   | <a href="#">C</a> , <a href="#">машинный код</a>   |
| <a href="#">Phalanger</a>                                              | Ms SS-PL[29] ( <a href="#">Shared source</a> )             | <a href="#">Microsoft IL</a>                       |
| <a href="#">Quercus</a> (в составе <a href="#">веб-сервера Resin</a> ) | <a href="#">GPL</a> или коммерческая                       | <a href="#">JVM</a>                                |
| <a href="#">PHC</a>                                                    | <a href="#">BSDL</a>                                       | <a href="#">C</a> , <a href="#">машинный код</a>   |
| <a href="#">Pipp</a> [30]                                              | <a href="#">Artistic License</a> и <a href="#">GNU GPL</a> | <a href="#">Parrot</a>                             |

На сайте [phpclasses.org](#) доступен сравнительный анализ некоторых компиляторов: [PHP compiler performance](#).

[править]Лицензирование

Основная статья: [Лицензия PHP](#)

[\[править\]](#)Приложения PHP

[\[править\]](#)Библиотека PEAR

Основная статья: [PEAR](#)

[\[править\]](#)Популярные веб-приложения

С использованием PHP разработано множество приложений, которые широко используются на различных [сайтах](#), [форумах](#) и [блогах](#).

[Drupal](#), [Typo3](#), [Mambo](#), [Joomla](#), [PHP-Nuke](#), [Wordpress](#) — [системы управления содержимым \(CMS\)](#);

[Magento](#), [osCommerce](#) — системы для Интернет-коммерции;

[Coppermine](#), [Gallery Project](#), [4images](#) — галереи изображений;

[MediaWiki](#), [DokuWiki](#) — вики-движки;

[phpBB](#), [SMF](#), [vBulletin](#), [Invision Power Board](#) — [форумные](#) движки;

[phpMyAdmin](#), [phpPgAdmin](#), [Adminer](#) — утилиты администрирования [СУБД](#);

[eyeOS](#) — удалённая [операционная система](#), основанная на принципе Desktop Operating System.

[\[править\]](#)Фреймворки

Для [быстрой разработки приложений](#) на PHP было создано множество [фреймворков](#), наиболее популярными из которых являются [Zend Framework](#), [CakePHP](#), [Symfony](#), [CodeIgniter](#), [Kohana](#) и [Yii](#)<sup>[31]</sup>.

[\[править\]](#)Интегрированные среды разработки для PHP

| Название                                      | Лицензия                                                                     | Сайт                                                                                              |
|-----------------------------------------------|------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| <a href="#">Komodo</a>                        | <a href="#">Mozilla tri-<br/>license (MPL/LGPL/<br/>GPL)</a> <sup>[32]</sup> | <a href="http://www.activestate.com">http://www.activestate.com</a>                               |
| <a href="#">JetBrains<br/>PhpStorm</a>        | <a href="#">Trial</a>                                                        | <a href="http://www.jetbrains.com/phpstorm/">http://www.jetbrains.com/phpstorm/</a>               |
| <a href="#">PHP<br/>Development<br/>Tools</a> | <a href="#">Eclipse Public<br/>License</a>                                   | <a href="http://www.eclipse.org/pdt/">http://www.eclipse.org/pdt/</a>                             |
| <a href="#">Zend<br/>Studio</a>               | <a href="#">Shareware</a>                                                    | <a href="http://www.zend.com/products/zend_studio/">http://www.zend.com/products/zend_studio/</a> |

|                                     |                                      |                                                                                                       |
|-------------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------------------|
| <a href="#">Aptana Studio</a>       | <a href="#">GNU GPL</a>              | <a href="http://www.apтана.org">http://www.apтана.org</a>                                             |
| <a href="#">phpDesigner</a>         | <a href="#">Shareware</a>            | <a href="http://www.mpsoftware.eu/">http://www.mpsoftware.eu/</a>                                     |
| <a href="#">PHP Expert Editor</a>   | <a href="#">Shareware[33]</a>        | <a href="http://www.phpexperteditor.com/">http://www.phpexperteditor.com/</a>                         |
| <a href="#">NetBeans IDE</a>        | <a href="#">CDDL</a>                 | <a href="http://www.netbeans.org/">http://www.netbeans.org/</a>                                       |
| <a href="#">RadPHP XE</a>           | <a href="#">Trial</a>                | <a href="http://www.embarcadero.com/products/radphp/">http://www.embarcadero.com/products/radphp/</a> |
| <a href="#">NuSphere</a>            | <a href="#">Trial</a>                | <a href="http://www.nusphere.com/">http://www.nusphere.com/</a>                                       |
| <a href="#">KDevelop[34]</a>        | <a href="#">GNU GPL</a>              | <a href="http://www.kdevelop.org/">http://www.kdevelop.org/</a>                                       |
| <a href="#">Microsoft WebMatrix</a> | <a href="#">Freeware</a>             | <a href="http://www.microsoft.com/web/webmatrix/">http://www.microsoft.com/web/webmatrix/</a>         |
| <a href="#">Notepad ++</a>          | <a href="#">GNU GPL</a>              | <a href="http://notepad-plus-plus.org/">http://notepad-plus-plus.org/</a>                             |
| <a href="#">Sublime Text 2</a>      | <a href="#">Proprietary software</a> | <a href="http://www.sublimetext.com/2/">http://www.sublimetext.com/2/</a>                             |

[править] Критика

[править] Несогласованный синтаксис функций и неортогональность

PHP предоставляет разработчикам большое количество самых разнообразных функций, которые попали в язык из расширений, создаваемых разными группами программистов. В результате синтаксис языка не согласован, например, часть функций для работы с массивами начинается с префикса `array_`, другая часть этим префиксом не обладает. Названия части строковых функций начинается с префикса `str`, другие функции таким префиксом не обладают. В тех же строковых функциях обрабатываемая строка может передаваться как в качестве первого, так и в качестве последнего аргумента, что вызывает путаницу у программистов, и, следовательно, требует постоянного обращения к документации. Некоторые задачи, например, разбиение строки на массив или подстроки, решаются несколькими функциями.

Другая причина отсутствия [ортогональности](#) в PHP заключается в особенностях появления этого языка на свет. В отличие от подавляющего большинства остальных языков программирования, PHP не был изначально задуман как полноценный язык программирования, а постепенно вырос

из [шаблонизатора](#) для [Perl](#). Это привело как к появлению в первоначальном варианте языка некоторых спорных решений, так и к отсутствию контроля со стороны создателя языка, который бы следил за стройностью его [архитектуры](#).

[\[править\]](#)Отсутствие обратной совместимости между версиями языка

Нижесказанное относится в большей степени к `php>=5.3.0`. Версии PHP < 5.3.0 сохраняет весьма хорошую обратную совместимость.

Код, созданный для более ранних версий языка, зачастую не работает или работает некорректно с более поздними версиями языка [\[35\]](#). В более поздних версиях исключаются конструкции, методики, функции, применявшиеся ранее. В результате, приложения, созданные несколько лет назад, практически теряют работоспособность для современных версий языка и требуют значительной модификации. Такие изменения обусловлены двумя факторами: устранением несогласованного синтаксиса и устранением конструкций, поощряющих создание небезопасного кода.

В новых версиях линейки 5.3.x большое количество функций было признано устаревшими, их поддержка не планируется в новых версиях языка [\[36\]](#), что вызывает несовместимость со скриптами, которые используют устаревшие функции. Также для версии 5.3 на данный момент отсутствует программное обеспечение Zend Optimizer. Однако разработчики планировали выпустить его в [2010 году](#)[\[37\]](#)

Надо отметить, что противоречие между обратной совместимостью и процессом развития — одна из ключевых проблем в разработке программного и аппаратного обеспечения. При работе над [скриптовыми языками](#) время от времени происходит резкая смена его [архитектуры](#) (а порой и [парадигмы](#)), обычно сопровождающаяся сменой первой цифры в номере версии. Так, в настоящее время идёт постепенный переход на новую ветвь языка Python — 3.x, в стадии тестирования находится [Perl 6](#), являющийся, по сути, новым perl-подобным языком. При этом принято выпускать переходные версии, в которых постепенно вводятся новые конструкции, а использование устаревших вызывает вывод предупреждений. К таким переходным версиям относится и PHP 5.3.

[\[править\]](#)Отсутствие поддержки многобайтовых кодировок в ядре языка

Поддержка [строк с многобайтовыми кодировками](#), такими как [UTF-8](#) реализуется через отдельные расширения mbstring и iconv, на уровне ядра поддержка отсутствует, однако с версии PHP 4.2.0 есть возможность переопределять стандартные функции работы со строками, подменяя их на аналоги из mbstring.

[\[править\]](#)Отсутствие поддержки многопоточности

В языке не предусмотрена возможность создания [многопоточных](#) приложений и отсутствует поддержка синхронизированного доступа к ресурсам, однако реализуема при помощи расширения PCNTL [\[38\]](#).

[\[править\]](#)Рынок труда

Согласно исследованиям одного из крупных [русских порталов трудоустройства](#)[\[39\]](#):

возрастной диапазон наиболее востребованных программистов PHP 20-45 лет; программисты PHP в возрасте до 30 лет составляют 78% от общего числа специалистов;

95% программистов PHP — мужчины;

80% программистов PHP владеют английским языком на базовом уровне и на уровне чтения технической документации; на разговорном и на свободном уровнях — 17%;

60% программистов PHP имеют высшее образование, 26% — неполное высшее, 8% — среднее специальное;

## Лекция 7. Услуги по созданию сайтов

Интернет-технологии ведения бизнеса. ЛР№1. HTML

Опубликовано Andrew 5 февраля 2012 в Программирование | Комментариев нет  
Нумерация лабораторных состоит из названия предмета, номера и названия работы.  
Данная лабораторная работа посвящена знакомству с HTML – языком разметки гипертекста.

### **1. Знакомство с HTML**

Ознакомиться с работой следующих тегов: html, head, title, body (bgcolor, background, text, link), p (align), b, em, u, i, font, img (см. ниже), ol/ul, li.

Обратить внимание на то, что теги бывают парные и непарные, а также на то, что у многих тегов имеются параметры.

**Задание 1.1.** Создать в Блокноте следующую web-страничку (Название в title указать “Знакомство с HTML”):

|                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Название страницы                                                                                                                                                                                     |
| Эпиграф                                                                                                                                                                                               |
| Кратко о себе несколько строчек. Описание должно содержать использование нескольких шрифтов, несколько слов выделить цветом. Страничка должна содержать все перечисленные теги с разными параметрами. |
| Число                                                                                                                                                                                                 |
| Что-то вроде подписи в виде графического файла                                                                                                                                                        |

*Работа с таблицами, рисунками, гиперссылками*

Для работы с таблицами используются теги table (border), tr (width, height), td/th (align, valign, width, colspan, rowspan)

**Задание 1.2.** Добавить к своей странице 2 таблицы с осмысленной информацией. Одна таблица – обычная, вторая должна содержать объединенные ячейки.

Для вставки рисунка используется тег img (alt, align, width, height, vspace, hspace, border).

**Задание 1.3.** Добавить к своей странице один рисунок три раза с разными размерами. Все используемые рисунки должны храниться в отдельной папке.

Гипертекстовые связи задаются при помощи тега a (href, target, name). Связи бывают трех видов: ссылки на ресурсы Internet, ссылка на сервис электронной почты, ссылки внутри документа.

**Задание 1.4.** Реализовать все три типа гиперссылок.

**Задание 1.5.** Сохранить работу на своем носителе. Работа понадобится в дальнейшем.

### **2. Работа с фреймами и картами изображений**

В настоящее время фреймы на практике **практически не используются**. Задание дано в основном для закрепления навыков работы с тегами.

**Задание 2.1.** Создайте рисунок, содержащий расположенные в разных местах названия сайтов. Названия должны располагаться в геометрических фигурах – круге, прямоугольнике, четырехугольнике, пятиугольнике. Организуйте переход на заданные сайты, используя карты изображений.

#### **Задание 2.2.**

Создайте web-страничку, состоящую из 3-х фреймов, названных условно “left”, “right”, “bottom” .

Фрейм “left” должен содержать перечень факультетов университета, при щелчке по названию факультета в фрейме “right” должно появиться описание факультета.

Во фрейм “bottom” поместить бегущую строку с приглашением поступать в университет. Бегущая строка должна быть разноцветной. *Для выполнения задания используйте пример из лекции.*

Создайте страничку состоящую из двух горизонтальных фреймов. В верхний фрейм необходимо поместить рисунок (или фотографию) с несколькими явно выраженными объектами. При щелчке по каждому объекту в нижнем фрейме должно появиться описание указанного объекта.

Выполненную работу сохраните на своем носителе.

### 3. Работа с формами

Для просмотра параметров формы можно изучить статью HTML. Работа с формами.

**Задание:** Составить регистрационную форму для проведения маркетингового исследования. Тему выберите самостоятельно. Форма должна иметь смысл. Необходимо использовать все элементы, рассмотренные на лекции.

В случае возникновения вопросов можно задавать вопросы в комментариях.

Каскадные таблицы стилей

Опубликовано Andrew 5 марта 2012 в Программирование | Комментариев нет



Выполните следующую лабораторную

#### работу по теме Каскадные таблицы стилей

Работа посвящена отработке навыков по настройке web-страниц и каскадных таблиц. Для работы используйте материалы с лекции. Посмотрите, как определить вид тегов HTML с использованием CSS, также посмотрите, как подключать внешний стилевой файл.

Выполните следующие задания.

2,5-мерный текст. Общая идея – три раза выводится один и тот же текст с небольшими смещениями, но с тремя разными цветами – красным, темнокрасным, серебристым. Тогда темнокрасный цвет создает тень, а серебристый цвет – блики.

Наберите пример из лекции.

Модифицируйте предыдущий пример, разбив содержимое предыдущего примера на два файла – первый должен содержать только информацию, второй – только описания стилей.

Создайте стилевой файл для HTML-документа из первой части первого задания. Можно использовать как внешний стилевой файл, так и определение стилей внутри html-файла.

По крайней мере опишите в стилевом файле следующие элементы для шрифта: название, размер, межстрочный интервал, размер красной строки.

На основании созданного описания для эпиграфа специально задайте зеленый цвет и отсутствие красной строки.

**Замечание.** Название свойств CSS для выполнения этой части задания я на лекции не давал. Предполагается, что Вы самостоятельно найдете информацию о свойствах в Интернет. Цель этого – понять, что все свойства знать нельзя, но всегда можно найти. Подключите статические фильтры для нескольких элементов на странице. Используйте Internet Explorer.

выведите рисунок тремя способами – обычный, перевернутый, инвертированный. примените к некоторому тексту эффекты glow и shadow

Попробуйте разобрать пример. Выведите дату с использованием рисунков и без использования рисунков. Во втором случае используйте данные комментария.



В данной лабораторной работе рассматривается возможность создания шаблонов с помощью **HTML** и **CSS**. Полученные навыки будут полезны при выполнении контрольной работы по HTML и CSS.

Трехколоночные макеты с использованием div

#### **Первый пример.**

В первом и втором примере использованы три колонки, где первая колонка задана в процентах, третья в пикселях, а на вторую отводится то, что осталось.

```
<html>
<head>
<style>
.header { background: #D5BAE4; }
.lll { position: absolute;
width:100%; padding:0px;
margin:0px}
.lll DIV { position: absolute; }
.col1 { background: #C7a3E4; width: 30%; }
.col2 { background: #EaDaC7; left: 30%; right: 200px; }
.col3 { background: #aCa5DE; right: 0; width: 200px; }
</style>
<title>Три колонки</title>
</head>
<body>
<div class="header">Шапка сайта</div>
<div class="lll">
<div class="col1">Колонка 1 задана в %</div>
<div class="col2">Колонка 2. Пример макета с
    тремя колонками, где первая колонка задана в %, третья в пикселях,
а вторая - то, что осталось.</div>
<div class="col3">Колонка 3 задана в px</div>
</div>
</body>
</html>
```

**Второй пример.** В данном примере опять же использованы три колонки, где первая и вторая колонка заданы в пикселях, а на вторую отводится то, что осталось.

```
<html>
```

```

<head>
<title>Макет с тремя колонками</title>
<style>
.header, .footer { background: #a5BAa4; }
.layout { overflow: hidden; }
.col1 { background: #C7b3E4; float: left; width: 200px; }
.col2 { background: #E0DbC7;
margin: 0 100px 0 200px; }
.col3 { background: #ECD5bE; width: 100px; float: right; }
</style>
</head>
<body>
<div class="header">Шапка сайта. Резиновая средняя колонка</div>
<div class="layout">
<div class="col1">Колонка 1 задана в пикселях</div>
<div class="col3">Колонка 3 задана в пикселях</div>
<div class="col2">Колонка 2<br><br>Подвал всегда ниже самой длинной колонки</div>
</div>
<div class="footer">Подвал</div>
</body>
</html>

```

**Задание.** Перенесите этот пример к себе и заполните осмысленной информацией.

Использование таблиц

Задание. Используя таблицы, попробуйте реализовать один из двух примеров с использованием таблиц.

Лабораторная работа. Знакомство с технологией Wiki

Опубликовано Andrew 26 марта 2012 в Internet | Комментариев нет



Технология wiki

С данной технологией Вы встречались, посещая сайт Википедии. В данной лабораторной работе рассматриваются основные команды на базе wiki-портала Сыктывкарского университета.

1. Зарегистрируйтесь на сайте **wiki.syktu.ru**. Для этого выберите “Представиться/зарегистрироваться”, а затем “Создайте учетную запись”. При этом в поле “Имя участника” укажите реальные фамилию и имя. Адрес электронной почты вводите по желанию.
2. После регистрации Вам будет предложено создать страницу. Создайте ее. После этого Вы попадете в окно редактирования страницы. В окне редактирования введите: [[Категория:Факультет управления 2012]] – это создаст категорию для Вашей страницы; Создайте категорию своей группы – Номер группы пробел Год, например “410 2012” Введите краткую информацию о себе, используя кнопки меню. Нажмите кнопку “Сохранить изменения”. Посмотрите, что получилось.
3. Для дальнейшего редактирования своей страницы выберите пункт “Править”. Добавьте следующие ссылки:  
на данное задание;  
на два Ваших любимых Интернет-ресурса, одна из ссылок должна быть выполнена с переименованием;

на трех однопользователей, используя внутренние ссылки и переименование;

Для этого используйте следующие команды:

[http://конкретный адрес] – для задания ссылки на внешний ресурс без переименования.

[http://конкретный адрес Альтернативное имя через пробел] – для задания ссылки на внешний ресурс с переименованием;

[[http://конкретный адрес в wiki-системе]] – для задания ссылки на внутренний ресурс без переименования (можно сделать ссылку на странички коллег);

[[http://конкретный адрес в wiki-системе|Альтернативное имя]] – для задания ссылки на внутренний ресурс с переименованием;

Нажмите кнопку “Сохранить изменения”.

4. Вставьте рисунок, желательно фотографию. Для этого необходимо предварительно загрузить рисунок через меню “Загрузить файл” в нижней левой части страницы.

Затем вписать строку

[[Файл:Название Вашего файл с точностью до регистра]].

5. Вставьте таблицу с неким осмысленным содержанием. Пример таблицы 2 на 2:

```
{|
|Ячейка 1*1
|Ячейка 2*1
|-
|Ячейка 1*2
|Ячейка 2*2
|}
```

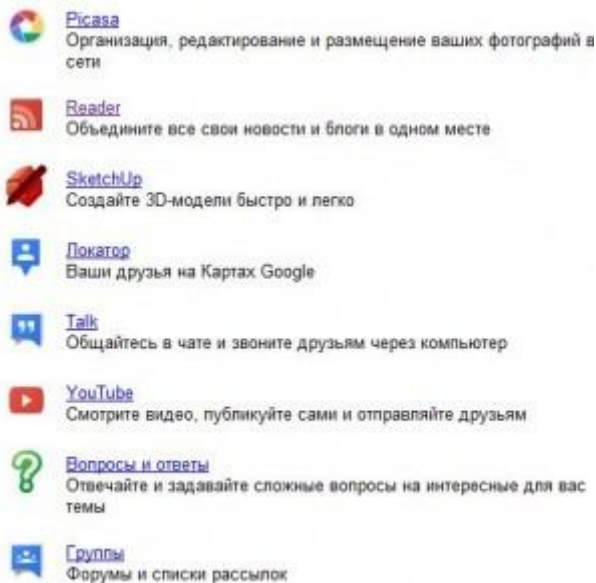
6. Создайте новую статью, набрав [[Название статьи]]. Наполнение статьи произвольное, но со смыслом.

7. На страничке, указанной преподавателем, напишите от своего имени хотя бы 2 предложения о том, как по Вашему мнению может использоваться wiki-система в учебном или другом процессе.

В результате выполнения этой работы, Вы должны создать 2 wiki-страницы (страница о себе и страница-сообщение) и оставить отзыв.

Лабораторная работа. Сервисы Google

Опубликовано Andrew 26 марта 2012 в Internet | Комментариев нет



Для выполнения данных заданий необходимо наличие аккаунта Google. Некоторые моменты создания аккаунта смотрите на странице Сервисы Google.

Лабораторная работа

## 1. Календарь Google

Используя меню Календарь аккаунта Google, распишите расписание дня на 2-3 дня вперед.

Откройте доступ к календарю для всех.

Добавьте на свою wiki-страницу ссылку на свой календарь.

## 2. Документы Google

Создайте документ, содержащий некое стихотворение. Откройте доступ к данному документу всем. Поместите ссылку на данный документ на своей wiki-странице.

Создайте документ, содержащий некий тост. Откройте доступ к данному документу 2-3-м коллегам. Поместите ссылку на данный документ на своей wiki-странице, указав кому открыт доступ. Для этого надо узнать электронные адреса тех, кому Вы хотите открыть доступ.

## 3. Документы Google. Формы

Используя вкладку Документы аккаунта Google, создайте новую форму, состоящую из 4-5 вопросов с различными вариантами ответов. Используйте какой-нибудь стиль.

Добавьте на свою wiki-страничку ссылки на редактирование и заполнение формы.

Добавьте 1-2 поля на свою страничку.

Заполните формы 2-х коллег.

Посмотрите результаты.

Добавьте на свою wiki-страницу скриншот таблицы с результатами формы (должно быть минимум 5 записей).

## 4. Документы Google. Таблицы

На wiki-страничке преподавателя перейдите по ссылке на **Ссылка для введения результатов**. Для того чтобы получить возможность редактирования данной странички, надо послать письмо на имя преподавателя с просьбой открыть доступ.

После получения доступа заполните данные таблицы.

Создайте документ, содержащий диаграмму. На wiki-странице разместите скриншот сделанной диаграммы.

5. Сервис **Вопросы и ответы**. Ознакомьтесь с указанным сервисом. Выберите понравившийся Вам вопрос, разместите ссылку на него на своей wiki-странице.

6. Познакомьтесь с рабочим столом от Google. Разместите на нем свои гаджеты.

Скриншот рабочего стола разместите на своей wiki-странице.

7. Сервисы хранения фотографий.

8. Социальная сеть Google+. Добавьте друг друга в сеть.

Создание сайтов с помощью CMS taba

Опубликовано Andrew 12 апреля 2012 в CMS | Комментариев нет



В настоящее время существует много online CMS. Примером такой CMS является taba. Познакомился с данным конструктором я примерно год назад. Мне он понравился своей простотой. В связи с этим я решил, что с данным проектом необходимо обязательно знакомить студентов в рамках дисциплины Internet-технологии ведения бизнеса.

Задание

На знакомство с **taba** я отвожу только одно лабораторное занятие. Необходимо сделать web-страницу, содержащую информацию о Вас. А именно:

Мне кажется, что все интуитивно понятно, но если не понятно, то задавайте вопросы.

Указать на главной странице фамилию, имя, группу, факультет.

Отметить, что это учебный проект. Сделать ссылку на это задание.

Добавить несколько новостей. Разрешить комментирование новостей.

Прокомментировать новости или иные записи на сайтах одноклассников. На своей странице завести страничку с ссылками на свою активность на других сайтах.

Обязательно добавьте ленту новостей какого-нибудь ресурса.

Добавьте несколько видео-файлов.

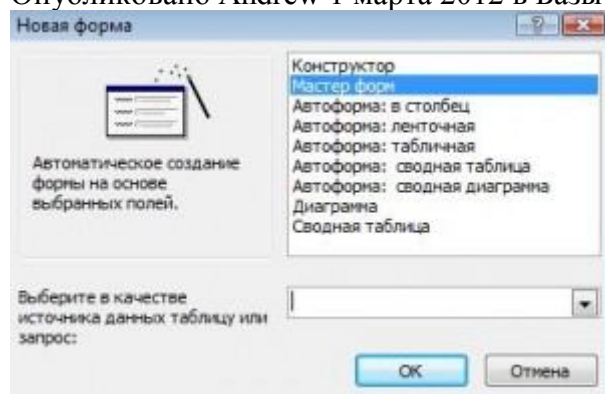
Добавьте форум с несколькими записями. Оставьте записи на форумах одноклассников.

Установите счетчик от LiveInternet.

Любые свои элементы.

Создание формы Access с помощью мастера

Опубликовано Andrew 1 марта 2012 в Базы данных | Комментариев нет



Назначение форм – облегчение ввода-вывода данных в таблицы, создание меню, командных кнопок, а также позволяет создавать различные меню.

Формы позволяют представить данные в наглядном и удобном виде. Для их создания могут использоваться встроенные *Мастеров по разработке форм*, а можно создавать непосредственно в конструкторах.

Для разработки форм на ленте **Создание** в разделе **Форма** имеется ряд полезных кнопок. При включение формы в левом верхнем углу окна Access на Ленте **Главная** имеется раздел **Режим**. Он позволяет выбрать один из трех возможных режимов работы с формой: **Конструктор**, **Режим Формы** и **Режим Макета**.

Рассмотрим подробно процесс создания формы “Покупатель”, предназначенной для ввода данных по покупателям. Для этого последовательно выбираем

Раздел *Форма* на Ленте **Создание** в окне базы данных.

В разделе *Форма* пункт **Другие формы**, в выпадающем списке выбираем **Мастер Форм**.

Поле *Выбор таблицы/запроса* для указания источника данных, по которым строится форма.

Переносим все *Доступные поля* в *Выбранные поля*. Нажмем кнопку *Далее*. Затем выберем внешний вид формы *в один столбец*, нажмем кнопку *Далее*. Затем выбираем стиль формы, задаем ей имя, сморим чтобы был выбран пункт *Открыть форму для просмотра и ввода данных* и нажимаем кнопку *Готово*.

В результате автоматически будет создана форма, используя которую можно вносить и просматривать записи. Отмечу, что перемещаться в форме можно или с использованием мыши, или клавиш Tab и Shift-Tab. Используя созданную форму введите данные по двум покупателям.

Имеются также *формы*: *Ленточная* и *Табличная*.

Задание. Создайте следующие формы для соответствующих таблиц.

“Покупатель” (описано выше). Внесите данные по 2 новым покупателям.

“Товар в ленточной форме” с помощью мастера *форма: ленточная*. Внесите дополнительно 2 описания товаров.

“Заказ”. Внесите данные по двум заказам.

Регистратор доменных имён

[править]

Материал из Википедии — свободной энциклопедии

Текущая версия страницы пока не проверялась опытными участниками и может значительно отличаться от версии, проверенной 23 ноября 2010; проверки требуют 33 правки.

Регистратор доменных имён — организация, имеющая полномочия создавать (регистрировать) новые доменные имена и продлевать срок действия уже существующих доменных имён в домене, для которого установлена обязательная регистрация. Таковыми доменами являются:

домен нулевого уровня (корневой домен);

все домены верхнего уровня (первого уровня);

некоторые домены второго уровня (например, `com.ru` или `co.uk`).

Во всех прочих доменах для создания поддоменов специальных полномочий не требуется.

Роль регистратора для корневого домена выполняет организация ICANN. Для многих доменов регистратор не единственный. При наличии нескольких регистраторов все они должны использовать единую (централизованную или распределённую) базу данных для исключения конфликтов и обеспечения уникальности доменного имени. Для того, чтобы стать регистратором доменов в

зонах `.com` `.net` `.org` `.biz` `.info` `.name` `.mobi` `.asia` `.aero` `.tel` `.travel` `.jobs`, необходимо получить аккредитацию ICANN[1]

В России с 4 декабря 1993 года до 1 июня 2001 года регистратор доменов в домене `.ru` был единственный, им являлся Российский научно-исследовательский Институт развития общественных сетей (РОСНИИРОС). После указанной даты регистраторов стало несколько (по состоянию на начало 2011 года — 25 регистраторов[2]), все они используют централизованную базу данных доменных имён. Единые правила для всех регистраторов в доменах `.ru`, и `.рф` задаются Координационным центром национального домена сети Интернет.

Во многих случаях регистратор доменных имён прямо или косвенно контролируется государством. Например, регистратором домена `.mil` является подразделение Министерства обороны США (Defense Information Systems Agency), а создание российского Координационного центра национального домена курировало Министерство связи РФ.

Регистрация и продление регистрации домена осуществляется в разных доменах на разных условиях — от бесплатной до весьма дорогой (до 10 тыс. долларов). Как правило, финансовые условия одинаковые для всех владельцев (администраторов) поддоменов в рамках одного домена.

Правила регистрации в международных (gTLD) доменах устанавливаются ICANN.

Правила регистрации в национальных (ccTLD) доменах устанавливаются их регистраторами и/или органами власти соответствующих стран. Эти правила могут быть разными, порой в них встречаются довольно своеобразные требования к владельцам доменов или условия владения доменом.

Технические функции регистратора доменов состоят в поддержании базы данных зарегистрированных доменов, предоставлении всем желающим доступа к этой базе по протоколу whois, а также в поддержании DNS-сервера (серверов) соответствующей зоны с NS-записями для всех зарегистрированных доменов. Поддержание других типов записей для зарегистрированных доменов, а также любых записей для их поддоменов не входит в функции регистратора — это отдельная услуга, её обычно оказывают иные компании, чаще всего интернет-провайдеры.

В ранние времена регистраторы доменных имён иногда принимали на себя функции по разрешению споров, связанных с доменными именами. Ныне почти все регистраторы такие споры рассматривать отказываются, отсылая стороны к судебным органам или

специализированным арбитражным органам (для UDRP). Сами регистраторы лишь исполняют судебные решения и выступают в суде в качестве эксперта или третьего лица. В связи со скандалом, связанным с действиями крупного регистратора RU-CENTER при регистрации доменов .РФ, Координационным центром была создана комиссия по спорным вопросам, связанным с аккредитацией[3], которая рассмотрела действия компании. 1 ноября 2011 года Арбитражный суд Москвы отменил решение ФАС, которая летом 2011 года признала RU-CENTER виновной в нарушении законодательства о защите конкуренции при старте регистрации доменов .РФ.[4][5]

## Domain registration

From Wikipedia, the free encyclopedia



This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. (September 2009)

Domain registration is the process of acquiring a domain name from a domain name registrar.

### Contents

[hide]

- 1 History
- 2 Designated registrar
- 3 DNS hosting
- 4 Domain name transfer
- 5 See also
- 6 References
- 7 External links
- 8 Sources

### [edit]History

In 1993, the U.S. Department of Commerce, in conjunction with several public and private entities, created InterNIC to maintain a central database that contains all the registered domain names and the associated IP addresses in the U.S. (other countries maintain their own NICs (Network Information Centers) -- there's a link below that discusses Canada's system, for example). Network Solutions, a member of InterNIC, was chosen to administer and maintain the growing number of Internet domain names and IP addresses. This central database is copied to Top Level Domain (TLD) servers around the world and creates the primary routing tables used by every computer that connects to the Internet.

Each ICANN-accredited registrar must pay a fixed fee of US\$4,000 plus a variable fee. The sum of variable registrar fees is intended to total US\$3.8 million.[citation needed] The competition created by the shared registration system enables end users to choose from many registrars offering a range of related services at varying prices.

### [edit]Designated registrar

Domain registration information is maintained by the domain name registries, which contract with domain registrars to provide registration services to the public. An end user selects a registrar to provide the registration service, and that registrar becomes the designated registrar for the domain chosen by the user.

Only the designated registrar may modify or delete information about domain names in a central registry database. It is not unusual for an end user to switch registrars, invoking a domain transfer process between the registrars involved, that is governed by specific domain name transfer policies.

When a registrar registers a com domain name for an end-user, it must pay a maximum annual fee of US\$7.34 to VeriSign, the registry operator for com, and a US\$0.18 annual administration fee to ICANN. Most domain registrars price their services and products to address both the

annual fees and the administration fees that must be paid to ICANN. Barriers to entry into the bulk registrar industry are high for new companies without an existing customer base.[citation needed]

Many registrars also offer registration through reseller affiliates. An end-user registers either directly with a registrar, or indirectly through one or more layers of resellers. As of 2010, the retail cost generally ranges from a low of about \$7.50 per year to about \$35 per year for a simple domain registration, although registrars often drop the price far lower – sometimes even free – when ordered with other products such as web hosting services.

The maximum period of registration for a domain name is 10 years. Some registrars offer longer periods of up to 100 years, but such offers involve the registrar renewing the registration for their customer; the 100-year registration would not be in the official registration database.

[edit]DNS hosting

Main articles: DNS hosting and Domain name system

Registration of a domain name establishes a set of Start of Authority (SOA) records in the DNS servers of the parent domain, indicating the IP address (or domain name) of DNS servers that are authoritative for the domain. This provides merely a reference for how to find the domain data – not the actual domain data.

Registration of a domain does not automatically imply the provision of DNS services for the registered domain. Most registrars do offer DNS hosting as an optional free service for domains registered through them. If DNS services are not offered, or the end-user opts out, the end-user is responsible for procuring or self-hosting DNS services. Without DNS services for the domain, the registration is essentially useless for Internet services, although this situation is often encountered with domain parking and cybersquatting.

[edit]Domain name transfer

A domain name transfer is the process of changing the designated registrar of a domain name. ICANN has defined a Policy on Transfer of Registrations between Registrars[1] The usual process of a domain name transfer is:

The end user verifies that the whois admin contact info is correct, particularly the email address; obtains the authentication code (EPP transfer code) from the old registrar, and removes any domain lock that has been placed on the registration. If the whois information had been out of date and is now updated, the end-user should wait 12-24 hours before proceeding further, to allow time for the updated data to propagate.

The end user contacts the new registrar with the wish to transfer the domain name to their service, and supplies the authentication code.

The gaining Registrar must obtain express authorization from either the Registered Name Holder or the Administrative Contact. A transfer may only proceed if confirmation of the transfer is received by the gaining Registrar from one of these contacts. The authorization must be made via a valid Standardized Form of Authorization, which may be sent e.g. by e-mail to the e-mail addresses listed in the WHOIS. The Registered Name Holder or the Administrative Contact must confirm the transfer. The new registrar starts electronically the transfer of the domain with the help of the authentication code (auth code).

The old registrar will contact the end user to confirm the authenticity of this request. The end user may have to take further action with the old registrar, such as returning to the online management tools, to re-iterate their desire to proceed, in order to expedite the transfer.

The old registrar will release authority to the new registrar.

The new registrar will notify the end user of transfer completion. The new registrar may have automatically copied over the domain server information, and everything on the website will continue to work as before. Otherwise, the domain server information will need to be updated with the new registrar.

After this process, the new registrar is the domain name's designated registrar. The process may take about five days. In some cases, the old registrar may intentionally delay the transfer as long

as allowable. After transfer, the domain cannot be transferred again for 60 days, except back to the previous registrar.

It is unwise to attempt to transfer a domain immediately before it expires. In some cases, a transfer can take up to 14 days, meaning that the transfer may not complete before the registration expires. This could result in loss of the domain name registration and failure of the transfer. To avoid this, end users should either transfer well before the expiration date, or renew the registration before attempting the transfer.[1]

If a domain registration expires, irrespective of the reason, it can be difficult, expensive, or impossible for the original owner to get it back. After the expiration date, the domain status often passes through several management phases, often for a period of months; usually it does not simply become generally available.[2]

The lists appear to all use at most 16 publicly available generic TLDs (gTLDs) that existed as of December 2009, plus .us. A February 2010 ICANN zone file access concept paper explains that most country code TLD (ccTLD) registries stopped providing zone files in 2003, citing abuse.

Домен верхнего уровня  
[править]

Материал из Википедии — свободной энциклопедии

Текущая версия страницы пока не проверялась опытными участниками и может значительно отличаться от версии, проверенной 18 июля 2012; проверки требуют 3 правки.

Домен верхнего (первого) уровня (англ. top-level domain — TLD) — в иерархии системы доменных имён (DNS) самый высокий уровень после корневого домена(англ. root domain). Является начальной точкой отсчёта (справа налево), с которой начинается доменное имя в Интернете.

http:// (www или домен(ы) n-го уровня) . домен 2-го уровня . домен верхнего (первого) уровня . /

например, http://www.wikipedia.org/

Поскольку каждое имя интернет-домена состоит из нескольких частей, разделённых точками и записанными в обратном порядке, то доменом верхнего уровня является завершающая из частей имени домена. Например, в имени домена ru.wikipedia.org доменом верхнего уровня является org, или ORG, поскольку в доменных именах регистр не имеет значения.

Содержание

[убрать]

1 Администрирование

2 Классификация

2.1 Основные домены

2.2 Общеупотребительные псевдо-домены

2.3 Устаревшие и неиспользуемые домены

2.4 Альтернативные и дополнительные домены верхнего уровня

3 Предлагаемые домены верхнего уровня

4 Численность доменов верхнего уровня

5 Примечания

6 Ссылки по теме

[править]Администрирование

Вопросами создания, поддержания и административного управления доменами верхнего уровня первоначально занималась руководимая Джоном

Постелом организацияIANA (от англ. Internet Assigned Numbers Authority —

«Администрация адресного пространства Интернет»), действовавшая на основании контракта с Министерством обороны США. После его смерти эти вопросы были переданы

в другую международную организацию ICANN — Интернет-корпорацию по присвоению имён и номеров (англ. Internet Corporation for Assigned Names and Numbers), а функции подрядчика перешли к Министерству торговли США.

В настоящее время ICANN обеспечивает поддержку и управление всем адресным пространством DNS в сети Интернет, кроме TLD ограниченного пользования, которые напрямую управляются американскими государственными организациями.

Технически домены верхнего уровня доступны через систему корневых серверов DNS, контролируемую ICANN.

С 12 января 2012 года, ICANN заявила об открытии свободной регистрации доменов первого уровня для всех интернет-пользователей.

[править]Классификация

[править]Основные домены

Общие домены верхнего уровня (англ. generic top-level domain — gTLD):

Неспонсируемые (основные) домены верхнего уровня.

Спонсируемые домены верхнего уровня (sTLD).

Домен верхнего уровня для инфраструктуры интернета.

Зарезервированные домены верхнего уровня.

Псевдо-домены верхнего уровня.

Национальные домены верхнего уровня (англ. country code top-level domain — ccTLD):

2-буквенный код страны (стандарт ISO 3166-1)

Домены страны на её языке (интернационализированный домен страны) (англ. IDN ccTLD).

[править]Общепотребительные псевдо-домены

См. также: Домены для примеров

Эти домены не присутствовали в адресном пространстве DNS, но они общепотребимы при пересылке почты из Интернета в сети с другим способом адресации. Для обработки писем, отправляемых на адреса в этом домене, почтовое программное обеспечение на конкретной машине, через которую отправляется почта, должно быть настроено соответствующим образом.

.uucp — для гейтования на машины, доступные при помощи UUCP.

.bitnet — для отправки почты в сеть BITNET.

.fidonet — для отправки почты в сеть Фидонет. В настоящее время, в связи с изменением общепринятой практики маршрутизации между сетями Интернет и Фидонет, вместо этого псевдодомена обычно используется домен .fidonet.org.

ряд старых систем так же использует домен верхнего уровня .local для адресов, применяемых в пределах одной машины или локальной компьютерной сети. Для адресации текущего компьютера так же, достаточно часто, применяется адрес .localdomain.

[править]Устаревшие и неиспользуемые домены

.nato — структуры международной организации НАТО — в настоящее время не используется, по крайней мере, в публично доступной части сети Интернет, откуда был удалён в июле 1996 года.

.web — домен, выделенный IANA для использования частным коммерческим регистратором Image Online Design. В связи с протестами общественности корневые серверы этого домена так и не были подключены к общей системе DNS. В настоящее время они продолжают функционировать, а на сайте регистратора находится сообщение о том, что он якобы проходит процедуру регистрации этого домена в ICANN.

.csnet — домен, предназначенный для связи с Computer Science Network — университетской и научной почтовой сетью в США. Перестал использоваться, по всей видимости, после объединения CSNET и BITNET, произошедшего в 1988 году.

.ddn — домен верхнего уровня, предназначенный для использования в американской оборонной сети Defence Data Network. Был запланирован, но так и не реализован.

[править]Альтернативные и дополнительные домены верхнего уровня

См. также: Альтернативные корневые серверы DNS

Теоретически кто угодно может установить и начать использовать свои собственные корневые серверы DNS. На практике в Интернет периодически появляются различные группы лиц и организации, которые открывают для публичного использования альтернативные корневые серверы DNS. Как правило, эти системы дополняют общепринятый набор доменов некоторым количеством новых доменов первого уровня, иногда — дополняют техническую реализацию. Например, до того, как DNS была расширена для возможности использовать в доменных именах символы национальных алфавитов, было предпринято несколько попыток создать дополнительные системы DNS, с доменными именами, в том числе первого уровня, содержащими символы того или иного национального алфавита, такие как русский домен я.ру. Эти попытки не получили широкого распространения, однако ряд таких проектов продолжает существовать и до сих пор.

Постольку, ICANN традиционно игнорирует альтернативные проекты, собственная деятельность этой организации по выдаче новых доменов верхнего уровня в своё время привела к конфликту вокруг домена .biz, на администрирование которого уже имелись два «исторических претендента». В результате этого ряд альтернативных систем DNS отказался распознавать домены, зарегистрированные в варианте ICANN .biz и полная совместимость их адресного пространства с DNS ICANN была потеряна.

Дополнительные домены верхнего уровня могут использоваться специализированным программным обеспечением, как правило — в пределах одного компьютера, для перехвата и последующей обработки части обращений к Интернет. Например, домен .onion используется анонимной сетью Тог для перехвата и последующей маршрутизации обращений к скрытым сервисам этой сети, а домен .i2p - программным обеспечением анонимной сети I2P.

[править] Предлагаемые домены верхнего уровня

См. также: Предлагаемые домены верхнего уровня (Proposed top-level domain) (англ.)

В данный момент рассматривается введение новых доменов верхнего уровня.[1][2] Абсолютно любые организации смогут регистрировать для себя свои собственные домены верхнего уровня, соответствующие различным торговым маркам, ассоциации — названиям отдельных целых отраслей, а власти — посвящать домены городам, регионам и т. п.[3]. По мнению экспертов, собственный домен первого уровня обеспечит безопасность бренда [4]. Домены начнут работу в 2013 году [5].

Человек, видящий интернет-адрес, должен точно определять язык его написания (на англ. The man who sees the Internet address should specify its language writing).

Среди первых кандидатов на получение домена первого уровня[6] — города мира, которые объединились в сообщество Dotcities. Туда входят Берлин, Нью-Йорк, Лондон, Вена и другие города мира.

24 августа 2010 года регистратор RU-CENTER выступил с инициативой создания доменов верхнего уровня .MOSCOW и .МОСКВА.[7] 30 сентября 2011 года проект поддержало Правительство Москвы.[8] Подача заявки планируется на 2012 год.

[править] Численность доменов верхнего уровня

На июль 2011 года:

доменов .com, .net, .org, .info, .biz соответственно: 95□871□283, 14□004□705, 9□266□204, 7□914□762, 2□106□138.[9]

русских доменов .рф, .ru и союзных .su: 858□734[10], 3□337□940[11], 93□883[12]. доменов .de: 14□500□342.[13]

Общая численность доменов верхнего уровня во всех доменных зонах в 2011 году превысила 225 млн имен.[14]

Самые крупные доменные зоны (по убыванию): COM, DE, NET, UK, ORG, CN, INFO, NL, RU, EU.[14]

## Подготовительный этап

Работа над сайтом должна начинаться с подготовительного этапа. Хорошая работа на этом этапе во-первых сэкономит вам много времени и сил в дальнейшем, а во-вторых «зарядит» вас информацией и идеями по теме сайта и позволит плавно перейти к основным этапам - подбору ключевых слов и написанию текстов.

Все работы подготовительного этапа надо проводить в «письменном виде». Записывать, перечитывать, править, добавлять, распечатывать, опять перечитывать и так далее. Мысли, перенесенные на бумагу, как бы освобождают место в голове для новых мыслей и идей и, кроме того, такое многократное перечитывание и переписывание само по себе рождает новые идеи.

Работа над сайтом, особенно написание текстов, требует большого объема ввода с клавиатуры, поэтому очень рекомендую вам обучиться слепой десятипальцевой печати. Я использовал и вам рекомендую программу VerseQ.

На подготовительном этапе надо:

Определить цель, с которой вы создаете сайт.

Определить позицию сайта.

Начать придумывать имя домена.

Провести предварительный сбор материала.

Определить список тем. Каждая тема - это, как минимум, одна HTML-страница.

Определить схему построения.

Составить план сайта.

Цель сайта

Формулировка цели должна быть простой, описывающей для чего вы хотите сделать сайт. Если цель сложная - лучше разбить ее на несколько простых, четко сформулированных целей.

Примеры целей:

Продавать с сайта продукт / услугу.

Привлекать посетителей для заработка на рекламе.

Делаю сайт просто так (у всех есть, хочу попробовать).

Считаю себя специалистом в некотором вопросе, хочу поделиться опытом.

Позиция сайта

О чем сайт (что будет на сайте). О машинах, о пластиковых окнах, о рыбках, о программировании и т.п. Для кого сайт. Для специалистов или для новичков, для молодых или для старых, для серьезных людей или для любителей приколов и т.п.

Чем будет отличаться от других подобных. Шириной охвата, узкой направленностью и большой подробностью, развеселым дизайном и содержанием и т.п.

Например:

Сайт о велосипедах.

Цель: Заработок на рекламе.

Позиция: Только детские 3-колесные велосипеды. Ориентируемся на родителей (а не на бабушек и дедушек), т.е. на молодых людей.

Отсюда вытекает:

Подойдут фотографии веселых карапузов, молодых красивых пар с детьми и т.п.

На сайте можно разместить материалы о детях, советы родителям, историю изобретения велосипеда.

На сайте можно разместить рекламу тур. агентств, детских товаров и т.п. Реклама сердечных капель наверняка будет лишней.

Рекламировать сайт надо тоже на молодежных ресурсах.

Короче говоря — хорошо продуманные и тщательно сформулированные цель и позиция сайта сразу же многое вам подскажут.

Имя домена

Придумать имя для вашего домена — ответственное дело с которым не надо торопиться. Оно вам понадобится, когда сайт будет готов. Тогда надо регистрировать домен и выбирать место под хостинг. Но начать подбирать имя нужно прямо сейчас. Подробнее о выборе имени домена можно прочитать в статье [Имя домена, whois, опасность ухода имен](#). Подробнее о регистрации и хостинге: [Регистрация домена](#), [Выбор хостинга](#).

Предварительный сбор материала

Перед началом работы надо провести предварительный сбор материала, найти полезные источники, оценить возможный объем. В некоторых случаях предварительный сбор материала уже проведен до начала работ над сайтом, или, даже до решения делать сайт. Можно считать предварительный сбор материала завершенным, если удастся сформулировать список тем и определить схему построения сайта.

Кроме того, сбор материала может происходить на любом этапе работ над сайтом, после окончания разработки сайта, в ходе его продвижения и поддержки. Добавление на сайт новых материалов и переработка старых — непрерывный процесс. Цель и позиция сайта должны служить ориентирами и границами при наполнении сайта, не давать уходить в сторону, задавать направление развития.

Список тем

Каждая тема — это, как минимум, одна HTML-страница. Некоторые темы могут разрастаться в несколько страниц или даже в большие подразделы сайта. По списку тем можно оценить минимальный размер планируемого сайта: число страниц больше или равно числу тем.

#### Схема построения

В общих чертах описывает на что будет похож сайт (книга, сборник статей, магазин, лента новостей) и как будут связаны страницы: цепочка, разделы/подразделы, общая куча и т.п. Так же хорошо разбить темы на основные, вспомогательные, справочные и т.п.

#### План

План должен описывать последовательность изложения материала. Точнее сказать, рекомендуемую вами последовательность ознакомления посетителя сайта с его содержанием. Посетитель может не придерживаться ваших рекомендаций и в соответствии со своими интересами ходить по ссылкам. Но вам все равно придется решить, что будет в начале главной страницы, что — в конце, ссылки на какие темы будут с главной страницы, как будет излагаться та или иная тема: на одной странице или на нескольких.

## Привлечение посетителей

Любой сайт, размещенный в Интернете, имеет одной из своих целей привлечение посетителей. Работу над их привлечением надо начинать еще на подготовительном этапе.

Посетители могут приходить тремя путями:

Естественный путь, вытекающий из самой идеи Интернета — переход по ссылкам с других сайтов. Этот путь естественный в том смысле, что нужен только Интернет (точнее — web) без поисковых систем, баннерных сетей и подобного. Можно добавить сюда и посетителей, приходящих с закладок в Избранном в своих браузерах. Главное на этом пути — интересный сайт на который другие люди захотят поставить ссылку.

С поисковых систем (ПС).

По рекламным объявлениям (баннерам).

Остановимся подробнее на первых двух пунктах. Работа по размещению рекламы делается после публикации сайта в Интернете.

### Естественный путь

Главное здесь — интересный сайт. Или как сейчас говорят — «хороший контент», т.е. хорошее содержание. Если ваши материалы интересны и уникальны, на них поставят ссылку, их с удовольствием разместят на своих страницах другие сайты, при этом по правилам хорошего тона тоже поставят ссылку на вас.

Интересным контентом может послужить все что угодно: новости техники, последние события в вашей индустрии, статьи на тему вашего бизнеса, статьи на тему ваших увлечений. Предоставляя своим посетителям полезную информацию вы получите в их лице постоянную аудиторию вашего сайта.

Кроме того, поисковые системы также любят сайты с богатым и разнообразным текстовым содержанием. Любовь ПС объясняется просто — они находят там много слов и фраз, встречающихся в запросах.

### Поисковые системы

Значительная часть посетителей может приходить на сайт с ПС. Чтобы таких посетителей было больше, надо сделать страницы сайта привлекательными для поисковых систем. Как это сделать? Рассмотрим общую схему поиска, каждому из вас хорошо знакомую.

#### Схема поиска:

Человек набирает в поисковой системе слова, информация по которым его интересует (поисковый запрос).

Поисковая система в своей базе данных ищет страницы, соответствующие запросу. Обычно это страницы, на которых встречаются слова из поискового запроса. Реже — страницы, на которые ведут ссылки, текст которых содержит искомые слова.

ПС сортирует ссылки на найденные страницы в соответствии со своим представлением о релевантности (степени соответствия) каждой страницы данному запросу.

Отсортированный список выводится порциями (экранами) по 10 ссылок и сопровождающей их краткой информации о странице (сниппетов).

Человек просматривает поисковую выдачу, обычно 1–2 страницы. Если находит подходящую ссылку — переходит по ней. Если нет — формулирует поисковый запрос иначе и повторяет с шага 1.

Нам надо как-то повлиять на конечный результат этого процесса — щелчок пользователя на ссылке. Как мы можем повлиять?

Пройдем еще раз по схеме поиска.

Поисковый запрос. Надо угадать или узнать какие поисковые запросы по теме (темам) нашего сайта делают люди. Как это сделать см. Технология подбора ключевых слов.

Поиск страниц, соответствующих запросу. Надо попасть в базу данных ПС. Для этого надо зарегистрироваться в поисковой системе (см. Поисковые системы, общее описание) и поместить на своих страницах слова из ожидаемых поисковых запросов (ключевые слова).

Сортировка страниц по ценности. Ценность страницы относительно поискового запроса складывается из двух составляющих:

Релевантность страницы по отношению к данному поисковому запросу. Предполагается, что если страница содержит информацию соответствующую поисковому запросу, в ее тексте слова из запроса будут встречаться чаще, будут выделены, будут входить в состав заголовков и т.п. Подробнее см. Оптимизация сайта для поисковых систем.

Работа по повышению релевантности по ключевым словам начинается на этапе проектирования сайта, главная часть работы выполняется на этапе написания текста и верстки. После публикации сайта, по мере необходимости, делается корректировка наборов ключевых слов и текстов по результатам анализа посещаемости.

Общий рейтинг страницы (или сайта), определяемый по числу ссылок с других сайтов. Предполагается, что на более полезные страницы и сайты будет больше ссылок с других сайтов. Работа над повышением общего рейтинга ведется постепенно и постоянно после публикации сайта в Интернете.

Вывод списка на экран. Тут ничего не поделаешь.

Человек просматривает список ссылок. ПС выводит ссылку на страницу и краткую информацию о странице, все это называется сниппет (snippet - отрезок, лоскут, обрывок). Первая строка сниппета, содержащая ссылку на страницу, выводится большими буквами жирным шрифтом и несколько строк шрифтом помельче.

В сниппет обычно попадает содержимое тега Title (реже - Description) и 2–3 строки текста страницы с найденными словами из запроса или первые 150–200 символов текста страницы.

Надо постараться, чтобы ваш сниппет привлек внимание, чтобы отводимые нам поисковой системой 3–5 строк заставили человека щелкнуть по ссылке. Лучше всего, если это будет хотя бы частично внятный текст, несущий информацию по теме запроса.

В середине августа 2006 AOL опубликовала «в академических целях» часть базы поисковых запросов. При анализе этой базы выяснилось, что примерно в половине случаев пользователь не переходит ни по одной ссылке из поисковой выдачи. Иными словами — результаты 50% поисковых запросов не кажутся подходящими. Одна из возможных причин этого — непривлекательные сниппеты.

Поисковые системы. Выводы.

Таким образом, для увеличения потока посетителей с ПС надо сделать следующее.

На этапе проектирования сайта:

Определить по каким поисковым запросам ваш сайт будут искать пользователи. См. Технология подбора ключевых слов.

Подобрать слова и фразы, соответствующие запросам (ключевые слова) См. Как оценивать ключевые слова. ч. 1 и ч. 2.

На этапе написания текста и верстки сайта:

Оптимизировать страницы сайта под эти ключевые слова. См. Оптимизация сайта для поисковых систем.

Оптимизировать страницы для поисковой выдачи, чтобы получались хорошие сниппеты. См. Поисковые системы, общее описание.

После публикации сайта в Интернете:

На основе анализа посещаемости периодически уточнять список ключевых слов и править тексты страниц.

Работать над увеличением ссылок с других сайтов (входящих ссылок). См. Продвижение сайта

Как устроены поисковые системы

Структура поисковых систем

Не вдаваясь в ненужные подробности, можно сказать, что ПС устроена таким образом

Паук (робот поисковой системы, crawler). Программа, которая ходит в Интернете по ссылкам. Период обхода может составлять от нескольких дней до нескольких недель. Отсюда запаздывание изменений в SERP по отношению к изменениям на сайте и в числе внешних ссылок.

Индекс. Список сайтов и страниц, которые обнаружил паук или были добавлены вручную. Ручное добавление сайта в индекс, видимо, будет сказываться в SERP после:

Очередного обхода паука, который найдет связанные страницы.

После очередного цикла работы подсистемы определения рейтинга. Рейтинг сейчас в большинстве поисковых систем рассчитывается на основе индекса цитирования, т.е. на основе внешних ссылок на сайт или страницу.

База данных. Хранилище информации о сайтах, которые обошел паук. Содержит тексты страниц и некоторую дополнительную информацию. Обновляется периодически во время обходов паука.

Система ранжирования сайтов. Состоит, по крайней, мере из двух подсистем:

Подсистема определения рейтинга страницы или сайта. Рейтинг определяется по количеству ссылок с других сайтов, причем ссылки с разных сайтов имеют разный вес. Эта система периодически пересчитывает рейтинги, используя информацию из БД. Периодичность может составлять несколько недель, отсюда запаздывание в SERP по отношению к изменениям в числе внешних ссылок.

Подсистема определения релевантности (соответствия) сайтов введенному запросу. Работает во время обработки запроса, на лету определяя релевантность сайтов из индекса на основании информации в БД.

Интерфейсная система. Принимает запросы и выводит результаты. Обычно ПС выводит содержимое тега title и несколько строк из текста страницы.

Снимок Интернета.

Из сказанного следует важный вывод: выдача поисковой системы формируются по некоторому «снимку» Интернета, запечатленному в базе данных ПС и может отставать от реального состояния Интернета на несколько дней или недель. В том числе и наши усилия по оптимизации проявятся не сразу и не на всех ПС одновременно.

Статистика запросов.

Для подбора хороших ключевых слов необходимо знать статистику запросов, сколько и каких запросов было сделано за определенный промежуток времени. Такой информацией обладают только сами ПС, некоторые из которых позволяют ей пользоваться через специальные сервисы, работающие следующим образом: пользователь вводит поисковую фразу; сервис сообщает ему количество запросов этой поисковой фразы за месяц. Некоторые сервисы выводят еще и список более узких поисковых фраз, содержащих данную, с числом запросов для каждой.

Например:

Окна 413000

Пластиковые окна 107137

Окна пвх 30363

Деревянные окна 19727

Дополнительные возможности поиска

Все ПС предоставляют при поиске дополнительные, более сложные инструменты, среди которых могут быть:

расширенный язык запросов;

географические сортировки;

сохраненные версии документов;

поиск в рамках сайта;

поиск похожих документов;

поиск через форму "расширенного поиска".

Однако посетители поисковых систем крайне редко используют расширенные возможности поиска. Обычно они меняют запрос и проводят новый поиск.

### Каталоги

Сайты ведущих поисковых систем содержат также и каталоги сайтов, рассортированные в разделы и подразделы по тематике. Внутри разделов сайты отсортированы по ссылочному рейтингу или по рейтингу посещаемости. Ссылки из таких каталогов учитываются при ранжировании страниц и имеют хороший вес. Некоторые ПС при прочих равных условиях отдадут предпочтение сайтам из своего каталога. Сайты помещаются в каталог вручную их авторами или SEO-специалистами.

### Общая тенденция.

Главное назначение поисковых систем — помочь обычному посетителю найти то, что ему нужно в Интернете. Обычный посетитель с этим вполне согласен и хочет, только чтобы поисковые системы делали это хорошо.

SEO-специалисты, и люди, продвигающие свои сайты, хотят от поисковых систем другого, чтобы они хорошо находили именно их сайты. И пытаются добиться этого с помощью различных приемов, некоторые из которых признаются разработчиками ПС допустимыми (белыми), а некоторые — недопустимыми (черными). На сайтах некоторых ПС даже публикуются рекомендации по "белой" оптимизации сайтов.

В силу этого поисковые системы развиваются одновременно в двух направлениях:

Совершенствуются алгоритмы оценки уважаемости и релевантности сайтов, определения тем, которым посвящен их контент. И, в конечном итоге, стараются, чтобы поисковая выдача давала посетителю то, что он ищет.

Совершенствуются алгоритмы «отстройки» от разного рода поискового спама, которым оптимизаторы пытаются воздействовать на поисковые системы. Некоторые из этих алгоритмов являются весьма сильнодействующими, как например, знаменитая «песочница» Google.

Кроме того, алгоритмы поиска и ранжирования полностью не раскрываются, чтобы оптимизаторам труднее было манипулировать результатами поисковой выдачи.

В сложных случаях (например, противоречивые советы по оптимизации) надо ориентироваться на интересы посетителя, так как поисковые системы развиваются именно в этом направлении.

### Технология подбора ключевых слов

Оптимизация страниц сайта по ключевым словам достаточно подробно описана в Интернете. Но сам подбор ключевых слов описывают или очень бегло («напрягите, как следует, мозги и

придумайте хорошие ключевые слова») или рекомендуют программы и on-line сервисы помогающие подобрать ключевые слова.

Отдельный случай, когда примерный набор ключевых слов задается изначально, например, при составлении ТЗ на разработку сайта. Но и в этом случае он не бывает исчерпывающим.

В любом случае стоит задача подобрать большое количество хороших ключевых слов.

Автоматизированные подборщики ключевых слов берут за основу предложенные вами ключевые слова или темы и подсказывают ключевые слова, отобранные по одному из следующих критериев:

Что еще искали с заданным ключевым словом, т.е. более узкие варианты запроса. Для запроса «окна» это будут «пластиковые окна», «окна пвх», «деревянные окна».

Что еще искали люди, делавшие запрос по вашему ключевому слову, т.е. по мнению поисковых систем, темы, связанные с вашей темой. Для запроса окна это будут стеклопакеты, рольставни, душевые кабины.

Какие слова встречаются на страницах, попавших в Топ (верхние строчки) поисковой выдачи по вашему ключевому слову.

Подборщики ключевых слов служат очень удобным вспомогательным инструментом, но имеют принципиальный недостаток: они могут определить, что искали в Интернете или писали на своих страницах другие люди, но они не знают, что хотите написать вы, куда вы можете повернуть свое повествование, какую еще тему затронуть. Это знаете только вы, да и то, может быть, не сразу, а узнаете в процессе работы на сайте. Это особенно актуально для тематического сайта, который делается не под заранее заказанный набор ключевых слов (семантическое ядро), а растет и развивается вместе с темой.

Кроме того, есть еще одна принципиальная трудность. Ключевые слова, которые приходят в голову, обычно имеют высокую конкуренцию (большое число страниц, которое находит ПС по этим словам) или малое число запросов, содержащих это слово, а нередко бывает, что и конкуренция высокая и запросов мало. Иными словами, мы думаем так, как думали до нас уже многие, или наоборот, погрузившись в свою тему, думаем не так, как другие люди, не погруженные в тему, например, наши потенциальные посетители. Получается, что придумать хорошее ключевое слово или фразу непросто. Частично эту проблему решают инструменты-подборщики ключевых слов, но далеко не на сто процентов.

Таким образом, предлагаемая в статьях по SEO (Search Engine Optimisation — оптимизация для поисковых машин) последовательность действий

Найдите ключевые слова.

Напишите под них текст.

Оптимизируйте HTML-код страницы.

Натыкается на серьезное препятствие в самом начале. Причина этого кроется в особенностях мышления, заклинивающегося на некотором наборе понятий, фраз, мыслей. Надо каким-либо образом разорвать этот порочный круг и дать доступ другим идеям и мыслям. Как это сделать?

Предлагаем простой способ.

Сначала пишем текст. Пишем текст нескольких основных страниц, одной, двух, трех — будет видно по ходу дела. Пишем, не думая о ключевых словах, думая о содержании: об удобствах, выгодах, особенностях вашего товара или услуги; о том, что вы знаете лучше других и чем хотите поделиться. Советы по написанию текста можно найти в статье Как писать текст. Писать надо подробно, тщательно, доходчиво, с примерами. Гладкость стиля на этом этапе не нужна.

Откладываем написанный текст на некоторое время, минимум на несколько дней, и занимаемся чем-нибудь другим.

Берем текст, написанный в пункте 1 и тщательно, вдумчиво читаем, анализируя на предмет ключевых слов. При этом поставьте себя на место человека, которого интересует информация на вашем сайте, и подумайте по каким ключевым словам он будет ее искать. Для расширения списка слов постарайтесь придумать необычные варианты запросов, т.е. представьте себе не только некоего усредненного человека, но и людей с пристрастиями или узко-специальными интересами. Если в написанном тексте не удастся найти хорошие ключевые слова, думаем, как можно изменить формулировки, что добавить, какой абзац написать подробнее, какая новая тема родственна затронутым и хорошо впишется в контент сайта. Каждое слово, подозрительное на роль ключевого, оцениваем (см. Как оценивать ключевые слова. ч. 1 и ч. 2) и записываем независимо от того, подошло оно или нет.

Распечатываем список ключевых слов (включая неподходящие) и время от времени перечитываем в любом мало-мальски подходящем месте: в транспорте, на прогулке с ребенком, в очереди. Приходящие в голову слова дописываем ручкой в список и проверяем.

По мере появления хороших ключевых слов, а они обязательно появятся, становится очевидным, что некоторые из них могут стать основой новой темы, новой страницы, раздела на сайте. Новые темы дадут новые ключевые слова и новые идеи для ключевых слов.

Повторяем шаги 3–5, пока не наберется нужное (с запасом) количество ключевых слов.

## Как оценивать ключевые слова

### ч. 1

Считаем, что у нас есть список ключевых слов (KW - Keyword). Наша задача оценить эти ключевые слова с точки зрения их способности привлекать посетителей с поисковых машин и выразить эту оценку в виде чисел. На всякий случай скажу: оценка в числовом виде позволит точно выстроить слова по рангу, чтобы плохие слова отбросить, лучшие слова использовать, средние слова оставить про запас.

Будем оценивать ключевые слова по количеству запросов в месяц и по числу страниц, которые находит поисковая машина по запросу, содержащему ключевые слова. Необходимое условие — поисковая машина и статистика запросов должны работать на одной базе данных. Для Рунета самая интересная пара Yandex — wordstat.yandex. Яндекс и его статистика запросов. Для англоязычного Интернета это связка Yahoo — Overture.

Дальнейшее изложение будет вестись на примере wordstat.Yandex — Yandex и может быть легко перенесено на случай Overture — Yahoo.

Число запросов, число показов, эффективное число запросов.

Строго говоря, Wordstat.Yandex показывает не статистику запросов, а статистику показов контекстной рекламы на странице (страницах) поисковой выдачи. Предполагается, что контекстная реклама показывается только на соответствующих ей страницах поисковой выдачи. Например, на страницах выдачи по запросу «окна» будет выводиться контекстная реклама по теме окна.

На самом Yandex написано: «Цифры рядом с каждым запросом в результатах подбора дают предварительный прогноз числа показов в месяц, которое вы получите, выбрав этот запрос в качестве ключевого слова».

Но нас интересует не число показов рекламы, а число запросов на поиск с этим ключевым словом. К счастью, Yandex дает статистику отдельных слов в поисковом запросе, надо только в настройках поиска включить соответствующую опцию. Например: если ввести запрос пластиковые окна, то мы увидим:

Статистика слов: пластиковые — 43 547 244, окна — 131 330 771

Поступим следующим образом:

Используем число запросов, которое дает поисковая машина Yandex по одиночным словам.

Используем число показов, которое дает Wordstat.Yandex для запросов, состоящих из одного слова.

Определим среднее отношение показов к запросам для запросов, состоящих из одного слова, и будем его использовать для определения числа запросов состоящих из нескольких слов. Назовем это вычисленное число эффективным числом запросов ( $Q_e$ ).

Все выше сказанное пояснено ниже на примере в Таблице 1. Дальнейшие вычисления будут проводиться с эффективным числом запросов  $Q_e$ .

Замечание 1

Возникает вопрос: Для каждого слова получается свое отношение показов к запросам, какую ошибку мы допускаем, используя среднее значение.

Надо учесть следующее:

Оценка ключевого слова в любом случае дает приближенный результат.

Оценка ключевого слова — это вспомогательный инструмент, который позволяет сказать, что одно ключевое слово лучше другого.

Если для двух ключевых слов мы получаем близкие оценки, то сказать заранее какое из них лучше невозможно — это будет определяться сложным набором неизвестных обстоятельств.

Таким образом, ошибка, получаемая при таком способе расчета числа запросов, будет вполне приемлемой с учетом общей неопределенности и приблизительности наших знаний об алгоритмах работы поисковых машин и поведении пользователей.

## Замечание 2

Надо учитывать, что для конкурентных ключевых слов число запросов, полученное на основании данных Wordstat.Yandex, будет завышено, так как большое количество SEO-специалистов мониторит выдачу по этим словам, вводя их день за днем в поисковую машину.

## Инструменты

Оценка и подбор ключевых слов требуют записи всех проверяемых слов, проведения некоторых однотипных арифметических операций и распечатки результатов. В первом приближении удобным инструментом послужит MS Excel — программа работы с электронными таблицами.

Надо создать электронную таблицу со следующей структурой:

Столбец 1 — Номер KW.

Столбец 2 — KW (Ключевое слово или фраза)

Столбец 3 — Np (число страниц).

Столбец 4 — S (число показов за месяц).

Столбец 5 — Q (число запросов за месяц).

Столбец 6 — SQ (S/Q).

Столбец 7 — Qe (Эффективное число запросов).

Столбец 8 — P (Qe/Np).

Потом добавим еще несколько столбцов с полезными характеристиками ключевых слов, вычисленными на основе описанных выше данных.

Правило заполнения ячеек в столбцах S, Q, SQ, Qe.

В столбец S записываем число показов в месяц из Wordstat.Yandex.

В столбец Q записываем или число запросов за месяц из Yandex, если запрос состоит из одного слова, или 0, т.к. Yandex не дает числа запросов по всей фразе из нескольких слов.

В столбце SQ пишем формулу =ЕСЛИ(Q>0; S/Q; 0). Вычисляем отношение числа показов к числу запросов, если число запросов известно.

В столбце Qe пишем формулу =ЕСЛИ(Q>0; Q; S/SQср). Т.е., если известно число запросов — берем их, если нет — пересчитываем показы в запросы. SQср - среднее значение SQ.

Правило работы со списком ключевых слов: заносим в список все слова, которые проверили, не зависимо от результатов оценки.

Для этого есть несколько причин:

Слово, которое пришло в голову один раз и не подошло, может прийти в голову еще раз и придется его опять проверять.

Слово в списке может натолкнуть на другое слово.

Проделанную работу надо по возможности сохранять.

Возьмем несколько ключевых слов на примере сайта, который мне пришлось оптимизировать и посмотрим, что получится.

Таблица 1. Исходные данные. Цветом выделены ключевые слова с близким значением P.

| N | Ключевое слово          | №<br>Число Стр. | S<br>Показы | Q<br>Запросы | SQ<br>Пок/Зап | Qe<br>Запросы Эф | P<br>ЗапЭф/Стр |
|---|-------------------------|-----------------|-------------|--------------|---------------|------------------|----------------|
| 1 | калькулятор             | 7 128 977       | 64 031      | 31 961       | 2.00          | 31 961           | 0.00448        |
| 2 | скачать<br>калькулятор  | 158 931         | 4 148       |              |               | 2 074            | 0.01305        |
| 3 | скачать<br>+калькулятор | 5 626           | 3 723       |              |               | 1 862            | 0.33087        |
| 4 | среднее<br>значение     | 485 092         | 819         |              |               | 410              | 0.00084        |
| 5 | среднее<br>отклонение   | 84 790          | 804         |              |               | 402              | 0.00474        |
| 6 | статистическое          | 3 551 401       | 73 206      | 15 033       | 4.87          | 15 033           | 0.00423        |
| 7 | дисперсия               | 248 271         | 10 211      | 3 076        | 3.32          | 3 076            | 0.01239        |
| 8 | дисперсия случайной     | 9 949           | 267         |              |               | 134              | 0.01342        |
| 9 | дисперсия<br>величины   | 14 023          | 269         |              |               | 13               | 0.00952        |

Считаем, что  $SQ_{ср} = 2$ . По данным приведенной таблицы получается другое число, но на самом деле набор ключевых слов содержал около 170 фраз, которые в среднем давали  $SQ_{ср}$  близкое к 2.

В таком представлении самая полезная колонка —  $P$ , отношение запросов к страницам. Чем больше число в этой колонке — тем лучше. Возникает вопрос: какое ключевое слово выбрать, если у нескольких слов близкие значения  $P$ ? Ответ во второй части.

Как оценивать ключевые слова

ч. 2

Как оценивать ключевые слова ч. 1

Какое ключевое слово выбрать, если у нескольких слов близкие значения  $P$

Рассмотрим ключевые фразы с близким значением  $P$ : номер 2 ( $P=0.01305$ ), 7 ( $P=0.01239$ ), 8 ( $P=0.01342$ ). Здравый смысл подсказывает, что лучше та фраза, у которой больше запросов ( $Q_e$ ). Чтобы облегчить восприятие, сведем данные из колонок  $P$  и  $Q_e$  в одну путем умножения:  $P * Q_e$ . Назовем это произведение  $KEI$ . Тогда слово с большим  $KEI$  — лучше. Поместим новую колонку  $KEI$  после колонки  $KW$ . Уберем колонки  $S$ (показы),  $Q$  (запросы) и  $SQ$  ( $S/Q$ ).

Таблица 1.  $KEI$  и исходные данные. Цветом выделены ключевые слова с близким значением  $P$  и хорошие ключевые слова.

| N | Ключевое слово       | KEI    | №<br>Число Стр. | Q <sub>e</sub><br>Запросы Эф | P<br>ЗапЭф/Стр |
|---|----------------------|--------|-----------------|------------------------------|----------------|
| 1 | калькулятор          | 143.29 | 7 128 977       | 31 961                       | 0.00448        |
| 2 | скачать калькулятор  | 27.07  | 158 931         | 2 074                        | 0.01305        |
| 3 | скачать +калькулятор | 615.92 | 5 626           | 1 862                        | 0.33087        |
| 4 | среднее значение     | 0.35   | 485 092         | 410                          | 0.00084        |
| 5 | среднее отклонение   | 1.91   | 84 790          | 402                          | 0.00474        |
| 6 | статистическое       | 63.63  | 3 551 401       | 15 033                       | 0.00423        |
| 7 | дисперсия            | 38.11  | 248 271         | 3 076                        | 0.01239        |
| 8 | дисперсия случайной  | 1.79   | 9 949           | 134                          | 0.01342        |
| 9 | дисперсия величины   | 1.27   | 14 023          | 135                          | 0.00952        |

KEI

Придуманное нами произведение  $P * Q_e = Q_e^2 / N_p$  носит название KEI (Keyword effectiveness index) и было введено в употребление Sumantra Roy (<http://www.1stsearchranking.com/>), известным SEO-специалистом. Он же предлагает такую градацию ключевых слов в зависимости от KEI:

KEI < 10 — бедные, нет смысла с ними работать

10 < KEI < 100 — хорошие

100 < KEI < 400 — очень хорошие

KEI > 400 — gifts (подарки)

Правда, формула предложенная Sumantra Roy немного отличается от нашей, в ней он использует Qd — число запросов за день (24 часа), умноженное на 1000.

$KEI = 1000 * Q_d^2 / N_p$ ,

Если считать, что в месяце 30 дней, то  $Q_d = Q_e / 30$ . Тогда получаем

$KEI = 1000 * Q_d^2 / N_p = 1000 * Q_e^2 / (900 * N_p) = 1.1 * Q_e^2 / N_p$ .

Получается, что KEI, рассчитанный по нашей формуле, занижен на 10 процентов (или, другими словами: имеет небольшой запас).

Вернемся теперь к выбору среднего SQ — отношения показов к запросам, используемого для расчета числа запросов, состоящих из нескольких слов. Можно действительно рассчитывать среднее значение SQ. Но это неудобно, т.к. при добавлении новых слов будет меняться SQ среднее, и соответственно  $Q_e$  и KEI у ранее введенных слов. Мы в своих расчетах приняли  $SQ_{ср} = 2$ .

Посмотрим, сильно ли мы ошиблись, что не взяли 1.5 или 2.5. Предположим, что правильное значение 2.5. В этом случае мы завысили KEI и приняли к рассмотрению несколько неудачных ключевых слов. С другой стороны, среднее SQ показывает, сколько страниц поисковой выдачи в среднем просматривает пользователь. Т.е., вместо 2 страниц пользователь просмотрит 2 с половиной, что увеличивает шансы нашего сайта. Таким образом, приняв SQ равным 2 вместо правильных 2.5 мы получим несколько слабых ключевых слов, шансы которых будут уравновешены более глубоким просмотром поисковой выдачи.

Аналогичные рассуждения, но с обратным знаком можно проделать для случая  $SQ_{ср} = 1.5$ . Не надо забывать, что все наши вычисления сугубо приблизительные и надо принимать во внимание еще и другие факторы.

KEI — не окончательный диагноз, а признак, позволяющий отсеять неперспективные слова. Дальше надо брать слова с KEI > 10 и рассматривать их более подробно.

Мы обычно еще используем:

Qd — число запросов за день.

KQ — Произведение KEI на Qd. Смысл этого показателя такой: говорим, например, что нас интересуют слова KEI > 20 и Qd > 100, получаем  $KQ_{min} = 2000$ . Дальше рассматриваем слова с  $KQ > 2000$ .

Величину  $P_1$ , обратную к  $P$  и округленную до целых:  $P_1 = 1/P = Np/Qe$ . Она удобна тем, что привычнее работать с целыми числами, а не с дробями. Приходилось читать, что имеет смысл работать со словами для которых  $P_1$  меньше 200.

Окончательно, наша таблица приобретает вид:

Таблица 2. Числовые характеристики ключевых слов. Цветом выделены хорошие ключевые слова.

| N | Ключевое слово       | KEI    | Qd<br>Qe в день | KQ<br>KEI*Qd | P1<br>Стр/ЗапЭф |
|---|----------------------|--------|-----------------|--------------|-----------------|
| 1 | калькулятор          | 143.29 | 1 065           | 152 656      | 223             |
| 2 | скачать калькулятор  | 27.07  | 69              | 1 871        | 77              |
| 3 | скачать +калькулятор | 615.92 | 62              | 38 218       | 3               |
| 4 | среднее значение     | 0.35   | 14              | 5            | 1 185           |
| 5 | среднее отклонение   | 1.91   | 13              | 26           | 211             |
| 6 | статистическое       | 63.63  | 501             | 31 887       | 236             |
| 7 | дисперсия            | 38.11  | 103             | 3 908        | 81              |
| 8 | дисперсия случайной  | 1.79   | 4               | 8            | 75              |
| 9 | дисперсия величины   | 1.27   | 4               | 6            | 105             |

Стоит отбросить слишком хорошие, слишком общие слова и слова с очень большим числом страниц в силу их высокой конкурентности и «нецелевого» характера. Остаются 3 фразы: «калькулятор», «скачать калькулятор», «дисперсия».

Слово «калькулятор» не проходит по параметру  $P_1 = 223$ , при рекомендуемом  $P_1 < 200$ . Но, во-первых границы приблизительные, а во-вторых слово очень подходящее для сайта, посвященного калькулятору.

Слово «дисперсия» имеет несколько значений и не всегда ищут именно статистическую дисперсию. Яндекс.Директ дает следующее числа:

Дисперсия 3658

Дисперсия пва 452

Акриловые дисперсии 158

Дисперсия формула 125

производство дисперсии 82

Статистика дисперсия 69

оценка дисперсии 57

расчет дисперсии 51

ветонит дисперсия 37

Дисперсия распределения 33

закупаем дисперсию пва 33

Таким образом, по этому слову может прийти некоторое количество нецелевых посетителей, которые ищут не вычисление дисперсии, а например, акриловую дисперсию. В одной из конференций мне пришлось встретить очень убедительное мнение по этому поводу: трафик нынче не слишком дорогой, а посетитель, если он пришел, может совершить полезные нам действия, а если он не пришел - то уж точно ничего полезного не сделает.

По числу запросов в день ( $Q_d$ ) можно оценить верхнюю границу дневного трафика:  $1065 + 69 + 103 = 1237$ . Конечно, не надо думать, что 1200 посетителей в день нам обеспечены, но если бы по нашим оценкам получалась верхняя граница 50 посетителей, это было бы очень плохо.

В дальнейшем наша задача будет состоять в том, чтобы реальное число посетителей было как можно ближе к нашей оценке.

Оптимизация сайта для поисковых систем.

(SEO - search engine optimization)

Главное назначение поисковых систем (ПС) — помочь обычному посетителю найти то, что ему нужно в Интернете. Обычный посетитель с этим вполне согласен и хочет только, чтобы поисковые машины делали это хорошо.

SEO-специалисты, и люди, продвигающие свои сайты, хотят от поисковых машин другого — чтобы они хорошо находили именно их сайты. И добиваются этого с помощью различных приемов, некоторые из которых признаются разработчиками ПС допустимыми (белыми), а некоторые — недопустимыми (черными). На сайтах некоторых поисковых систем даже публикуются рекомендации по правильной оптимизации сайтов.

В силу этого поисковые машины развиваются одновременно в двух направлениях:

Совершенствуются алгоритмы оценки уважаемости и релевантности сайтов, определения тем, которым посвящен их контент.

Совершенствуются алгоритмы «отстройки» от разного рода поискового спама, которым оптимизаторы пытаются воздействовать на поисковые машины. Для поискового спама или спама ПС придуман даже специальный термин — спамдексинг.

Кроме того, алгоритмы поиска и ранжирования полностью не раскрываются, чтобы оптимизаторам труднее было манипулировать результатами поисковой выдачи. В сложных случаях (напр.: противоречивые советы) SEO-специалистам надо ориентироваться на интересы посетителя.

Главная идея поисковой оптимизации (SEO)

Главная идея оптимизации состоит в следующем: для каждой (по возможности) страницы сайта надо сделать так, чтобы страница была расценена ПС как релевантная поисковому запросу и попадала в первые 10–30–50 позиций (Тор 10 — Тор 30 — Тор 50 ).

Из этого простого соображения следуют выводы:

Надо знать, какие запросы нас интересуют. Это в свою очередь определяется темой страницы.

Здравый смысл подсказывает, что надо выбрать небольшой перечень запросов для отдельно взятой страницы.

Надо оценить достижимость нашей цели, учитывая конкуренцию за места в поисковой выдаче. И, может быть, для начала выбрать менее конкурентный запрос, иначе сформулировав сам запрос или увеличив число слов в запросе (например: не «пластиковые окна», а «пластиковые окна установка»).

Надо знать, как ПС определяет релевантные страницы и сделать страницу в соответствии с этими правилами.

Надо знать, что ПС считают поисковым спамом и не делать этого.

Как ПС определяет релевантные страницы.

Основываясь на рекомендациях самих ПС и опыте SEO-специалистов, можно сформулировать ряд утверждений.

Поисковые системы работают только с текстом. В том числе и поиск изображений (например: Яндекс.Картинки) работает с подписями к изображениям, именами файлов, текстом рядом с картинкой, а не с изображением.

ПС считает, что страница отвечает поисковому запросу, если в ней присутствуют слова из запроса (ключевые слова, KW) и выполняется еще ряд условий:

KW встречаются чаще чем обычно.

Играют особую роль (см. ниже).

Ключевые слова расположены характерным образом: близко к началу текста и близко к концу. Такое расположение слов, характеризующих тему страницы соответствует естественному расположению слов в текстах: в начале текста мы пишем, что хотим рассказать о том-то и том-то, используя характерные для этой темы слова, а в конце текста делаем выводы, снова используя слова из этой темы.

ПС считает, что страница отвечает поисковому запросу, если на нее ведут ссылки в тексте которых присутствуют ключевые слова.

Поисковые системы накопили большой объем статистики касающейся текстов, их тематики, структуры, лексики и т.п. Основываясь на этих данных, ПС совершенствуют алгоритмы определения «естественности» текста и все более успешно отсеивают тексты созданные специально для поисковых систем, а не для посетителей сайта.

Как повысить роль ключевых слов в тексте

Надо поместить ключевые слова туда, на что ПС обращает больше внимания, и — соответственно, чему придает больший вес.

Тег <title> — заголовок страницы, отображается в заголовке окна браузера. Значение этого тега увеличивается еще и тем, что его содержимое выводится как заголовок сниппета в SERP и прочитывается человеком, который ввел запрос. Если заголовок понравится — человек щелкнет по ссылке, если нет — то не щелкнет.

Тег <meta description> — описание страницы. Так же может выводиться в сниппет и влиять на выбор ссылки.

Тег <meta keywords> — собственно список ключевых слов страницы. Ввиду неумеренного спама этого тега в прошлом, современные ПС не обращают на него внимания. Но поместить туда ключевые слова не повредит.

<h1> — самый главный заголовок текста страницы. Должен быть один.

<h2> — <h6> — менее важные заголовки.

<b>, <strong> — выделение отдельных слов. Не смотря на то, что визуальный эффект этих тегов один и тот же — выделение жирным шрифтом, смысл у них разный. <b> — тег визуальной разметки, а <strong> — тег структурной разметки, предписывающий сделать на выделенном слове или фразе логическое ударение (например, для речевого браузера).

<a> Тексты ссылок. Если мы ссылаемся на родственную по тематике (судя по наличию KW в тексте ссылки) страницу, это хорошо. А если это внутренняя ссылка, то еще лучше.

<img Alt="", Title=""> — подписи к картинкам.

Названия файлов страниц и картинок. В качестве разделителя слов лучше всего использовать «-» (минус).

Использовать в тексте синонимы ключевых слов и слова из близких тем. Например, если ключевые слова «пила, пилить», то упомянуть в тексте «молоток», «инструмент» и т.п.

Что может быть отнесено к поисковому спаму

Нет исчерпывающего определения поискового спама или четких границ, отделяющих допустимые приемы от спама. Вот что пишет по этому поводу Yandex:

«Поисковый спам» — это попытки обмана поисковой системы и манипулирования ее результатами с целью завышения позиции сайтов (страниц) в результатах поиска. Сайты, содержащие «поисковый спам», могут быть понижены при ранжировании или исключены из поиска Яндекса из-за невозможности их корректного ранжирования.

Мы не комментируем исключение сайтов из поиска и/или изменение их позиции, не уведомляем о них и не даем никаких гарантий и сроков на повторное их включение в поиск. Мы также не сообщаем исчерпывающий список методов спама, которые могут вызвать исключение сайта из поиска и/или изменение их позиции.

Источник: Лицензия на использование поисковой системы Яндекс (Yandex.ru)

Мы стараемся не индексировать или не ранжировать высоко:

автоматически сгенерированный (бессмысленный) текст,

сайты, копирующие информацию с других ресурсов и не создающие оригинального контента или сервиса,

сайты, представляющие товары или информацию по партнерским программам, но не создающие для пользователя никакой добавленной ценности,

страницы с невидимым или слабовидимым текстом или ссылками,

сайты, отдающие разный контент пользователю и роботам поисковых систем («клоакинг»),

страницы и сайты, единственной целью которых является перенаправление пользователя на другой ресурс, автоматически («редирект») или добровольно (реклама без контента),

сайты, на которых из-за обилия навигационных меню, новостей, вспомогательной информации, рекламы трудно найти их основное содержание,

страницы и сайты, ставящие ссылки исключительно для обмана поисковых систем и накачивания релевантности,

сайты или группы сайтов, интенсивно ссылающиеся друг на друга («линкфармы»),

ссылки, не являющиеся рекомендацией автора посетить ресурс.

Источник: Чем отличается качественный сайт от некачественного с точки зрения Яндекса?

Таким образом, оптимизация сайта для поисковых систем заключается в выполнении простых правил и главное — не перегнуть палку. Простым критерием может служить «читабельность» текста, если текст нормально читается — значит все нормально и с точки зрения человека и с точки зрения поисковой системы.

## Как писать текст

Написание текстов дело не простое и требует некоторых способностей. Но, если учесть, что наша цель — просто написать толковый текст для сайта, то задача сильно упрощается. Кроме того, есть несколько приемов, позволяющих писать приличные тексты. Надо только хорошо знать то, о чем пишешь и следовать приведенным ниже советам.

### Цель

Определите с какой целью и для кого будет написан текст. Чего вы хотите добиться своим текстом — заставить посетителя оставить свои контактные данные, подписаться на рассылку, купить, перейти по ссылке или просто сообщить информацию. Хорошо продуманная и тщательно сформулированная цель поможет вам писать.

### Сбор информации

Прочтите на заданную тему все, что сможете найти: книги, газеты, журналы, поищите в Интернете, подпишитесь на подходящие рассылки и посмотрите их архивы, поговорите со знающими

людьми. Сохраняйте все добытые сведения: записывайте на бумаге, собирайте вырезки, делайте ксерокопии, сохраняйте в компьютере электронные документы.

### Обработка

Следующий шаг — обработка полученной информации. Просмотрите все собранные материалы, избавьтесь от дубликатов, рассортируйте, отфильтруйте лишнее. Проверьте материал на полноту, если чего-то не хватает — поищите еще.

### План или краткий конспект

Продумайте схему изложения. Напишите план или краткий конспект. Организуйте материал в логическом порядке, в соответствии с продуманной схемой изложения.

Оставшаяся работа будет состоять в раскрытии каждого пункта плана. Шаг за шагом, раскрывая пункты плана, вы напишите все, что хотели и не пропустите чего-либо важного.

### Первый вариант

Все, что вы пишете, должно иметь логическое начало, основную часть, и заключение. Написав план, вы как раз и приготовились к последовательному и логичному изложению материала. Хорошим началом является описание некоторой проблемы (пусть даже маленькой), тогда основная часть будет посвящена решению этой проблемы, а заключение — выводам.

Пишите так, словно обращаетесь к одному человеку. В сущности, подготовив материал, вы свели работу к обсуждению проблемы с читателем. Представьте себе, что инструктируете человека, который хочет выполнить определенную работу, но не имеет представления о том с чего начать и как довести дело до конца.

Начав писать, не останавливайтесь, пока не завершите работу. Для первого варианта текста не важна гладкость изложения и правильный подбор слов, если затрудняетесь с выбором слова — пишите варианты через дробь или в скобках. Пишите, не останавливаясь, следуя за развитием мысли.

Поставьте себе задачу написать, как минимум, 500 слов. Во-первых, поисковые машины любят страницы, где много текста (см. статью Поисковые системы, общее описание). Во-вторых, 500 слов вы потом легко превратите в 700–800. Увеличивать объем имеющегося текста просто: здесь добавили пару слов, там добавили предложение, где-то в другом месте более подробно написали абзац.

Очень удобно перед заголовком текста, самой первой строкой написать: XXX слов, время от времени пересчитывать и записывать число слов. Приятно наблюдать, как растет объем текста. В Word это сделать просто: Сервис — Статистика.

### Перерыв

Окончив, сделайте перерыв в работе, переключитесь на что-нибудь другое. Смысл этого в том, чтобы отложить работу на несколько дней, дать себе хорошенько отвлечься и потом взглянуть на свой текст свежим взглядом.

### Второй вариант

Через несколько дней возьмите написанный вами текст и прочитайте его критически (см. статью Каким должен быть текст). Поправьте неудачные предложения, подберите нужные слова и более удачные формулировки. Удостоверьтесь в том, что:

вы написали то, что хотели;

ваш текст легко читать;

ваш текст легко понять;

ваш текст имеет законченную логическую структуру.

Последующие варианты

На текст можно посмотреть, как минимум, с двух точек зрения. С точки зрения производителя (автора, продавца, учителя) и с точки зрения потребителя (читателя, покупателя, ученика).

Потребитель может доброжелательным, таким, который хочет разобраться, купить, понять и готов простить мелкие огрехи и нестыковки, если материал добротный и толковый. А может быть недоброжелательным, обращающим главное внимание на мелкие недоделки и недостатки.

Когда вы писали текст, вы писали его с точки зрения производителя. Постарайтесь теперь взглянуть на него глазами потребителя. Взгляните на ваш текст как доброжелательный потребитель — все ли понятно, логично, подробно. И, как недоброжелательный потребитель — все ли аккуратно, точно, правильно стыкуется.

Поставьте себя на место потребителя и подумайте, какие дополнительные вопросы он может задать. Постарайтесь ответить на эти вопросы и сделайте дополнения.

Перечитывайте и переписывайте ваш текст, пока не почувствуете, что лучше уже не будет. Очень полезно распечатать текст и прочитать его с листа бумаги. Почему-то в распечатанном тексте бросается в глаза то, что осталось незамеченным на дисплее.

Одновременно старайтесь:

Расширить текст, добавляя новые мысли, подробнее поясняя, убедительнее аргументируя.

Сократить текст, убирая повторы, лишние слова, упрощая тяжеловесные обороты.

Пусть ваш текст приобретет стройность, лаконичность, динамизм, информационную насыщенность.

Как писать привлекающие внимание и хорошо читаемые тексты

Чтобы достичь цели, поставленной при написании текста, необходимо завоевать доверие читателя, расположить к себе. Своим текстом вы должны показать, что вы надежный, квалифицированный, грамотный человек и вашим словам можно верить.

Но прежде всего — с заголовка, с первой фразы текст должен увлекать и заставлять читать дальше. Надо сразу захватить внимание: сделать выгодное предложение, обнаружить проблему и заинтриговать, задать вопрос. Дальше надо убеждать читателя описанием преимуществ вашего

товара, интересными подробностями, качеством сообщаемой информации. И обязательно — ваш текст должен легко читаться.

#### Размер текста

Объем информации, который человек в состоянии воспринять «с налета», без предварительной подготовки, не велик. По-этому, текст на странице не должен быть слишком длинным. Почти любую тему (хотя бы в общих чертах) можно уложить в тысячу, полторы тысячи слов. Если текст разрастается больше чем на 1500 слов — в нем либо начинают обособливаться логические части и его надо разделить, либо он страдает многословием и его надо сократить.

С другой стороны, считается, что поисковые машины любят страницы с большим количеством текста, и нижняя граница здесь примерно 500 слов.

Конечно, 500 и 1500 слов — приблизительные границы. Кроме того, поисковые машины игнорируют предлоги, союзы и другие, так называемые «стоп слова», которые в русских текстах составляют от 10 до 20 процентов текста.

#### Составляющие текста.

Резюме. Его цель — позволить читателю быстро ознакомиться с основными мыслями и фактами текста. При этом чтобы понять резюме, не обязательно заглядывать в сам текст. Явно разделите основной текст и резюме.

Некоторые новостные порталы в начале своих статей помещают резюме, не отделяя его явно от текста. После прочтения резюме, при переходе к основному тексту возникает неприятное впечатление: только-что это прочитал, а тебя заставляют прочитать еще раз.

Введение. Хорошо, если введение говорит о том, почему надо прочитать сам текст: что нового и полезного узнает читатель. Во введении можно сформулировать проблему или чем-нибудь озадачить, а в основной части решить проблему, объяснить, в чем суть дела.

Заключение. Здесь можно сделать выводы, кратко повторить сказанное, усилить акценты, повторно озвучить основную мысль, т.е. в ненавязчивой форме «усилить и углубить».

Автор, версия, дата. Если это уместно, укажите имя и координаты автора. Иногда бывает полезно ставить версию документа. Укажите дату написания текста, а если он описывает действия или события — и их дату тоже.

Хорошо указать дату написания документа и дату последнего изменения. Если документ написан несколько лет назад, а дата изменения недавняя — это говорит о том, что текст не брошен на произвол судьбы, над ним работают, он содержит проверенную и в тоже время свежую информацию.

#### Восприятие текста

Текст должен легко читаться. Основное правило: избежать большой однородной массы текста, при взгляде на которую пропадает желание читать. Кроме того, чтение с монитора утомляет глаза. Используйте пустое место, что бы дать отдых глазам посетителя.

Используйте многоуровневые заголовки. Как показывают исследования, web-пользователи сканируют страницу, пока взгляд не остановится на цепляющих его словах. Как раз для

привлечения внимания служат заголовки и подзаголовки — они подскажут, о чем текст, послужат кратким оглавлением страницы.

Разбивайте текст на абзацы. Возьмите за правило: отдельная мысль — отдельный абзац.

Используйте в качестве разделителей пустые строки.

Используйте списки. Любое перечисление лучше организовать с помощью списка, а не с помощью запятых.

Используйте шрифтовые выделения. Жирность, цвет, разрядка, цвет фона.

Текст должен легко пониматься. Подбирайте точные выражения. Пишите кратко. Подбирайте слова и термины, которые будут однозначно поняты вами и теми, кто будет читать.

Уберите из текста все лишнее: ненужные повторы, очевидные, общеупотребительны фразы. Например: «мы предлагаем широкий выбор товаров по низким ценам». Надо или убрать эту фразу или подтвердить цифрами и фактами: «мы предлагаем 456 наименование товаров с 10% сезонной скидкой»

Сформулируйте иначе то, что может быть непонятно: объяснения через неизвестное, неточные определения.

Пишите конкретно. Слова «мало», «много», «часто», «очень» и подобные дают лишь иллюзию уточнения. Везде, где возможно, указывайте точные значения, т.е. числа. Вместо «широкий выбор» — «456 наименований», вместо «низкие цены» — «на 10% ниже рыночной».

Избегайте штампов, бюрократических выражений, витиеватых оборотов.

Пишите коротко. Короткими предложениями и короткими абзацами. Придерживайтесь правила «один абзац — одна мысль».

Заменяйте иностранные слова русскими.

Замечание. Повторы не всегда плохо. Без них не обойтись, если вы хотите подчеркнуть главную мысль или обратить внимание на важное обстоятельство.

Еще пара советов с сайта intranetdesign.ru — Как писать деловые тексты

Пишите в активном залоге. Деловой стиль не предполагает обезличенность — за текстом должен быть виден человек, а не мифическое руководство или соображения необходимости. Пассив скрывает информацию, поэтому его часто используют, чтобы скрыть ответственность. Актив подталкивает к полному изложению. Например, во фразе «Отдел персонала закончил проект» активная форма глагола требует указания того, кто выполнил работу. А фраза «проект выполнен» скрывает эту информацию.

Чтобы сделать текст живым, замените отглагольные существительные глаголами. Например, фраза «Мы договорились встретиться» лучше, чем «Договоренность о встрече была подтверждена».

В руководствах к действию описывайте конкретные шаги. Исследования в социальной психологии показывают, что многие люди не выполняют требуемые действия, если не знают четкой

последовательности своих шагов. Это происходит потому что людям надо сразу знать сколько сил и времени на это потребуется. За помощью многие обращаются неохотно, т.к. стесняются задавать «глупые» вопросы или испытывают сложности в общении.

Если у человека возникла задача, он должен прочитать инструкцию, ее решить. Поэтому инструкция должна быть написана исходя из полного решения задачи человека, а не из потребностей одного отдела, который обеспечивает лишь часть решения.

Пример:

Для получения ссуды Напишите заявление (образец)

Завизируйте заявление у непосредственного руководителя

Передайте заявление директору по персоналу Ивановой Алле за 2 недели до желаемого срока получения ссуды. Заявления рассмотрит генеральный директор. О решении вам сообщит Иванова Алла.

За деньгами обратитесь в кассу, она открыта ежедневно с 9.00 до 11.00. Отдайте кассиру свое заявление.

Такое описание лучше, чем:

Каждый сотрудник нашей компании может получить ссуду. Заявление за подписью своего руководителя подавайте в отдел персонала.

Структура текста

От главного к второстепенному. Старайтесь, чтобы развитие темы шло от общего к частному. Это позволит читателю быстро понять содержит ли раздел нужную ему информацию. В предложении первые слова описывают главное действие или событие, следующие — подробности и обстоятельства. Первое предложение абзаца описывает основную мысль, следующие предложения — уточняют и раскрывают.

Заголовки делают очевидной структуру текста. Они должны раскрывать содержание разделов, быть точными, краткими и как можно более конкретными. Используйте не более 2–3 уровней подзаголовков. Если внутри раздела несколько самостоятельных тем, их лучше визуально разделить пустыми строками.

Опорные слова — это словосочетания, раскрывающие смысл абзаца или предложения. Они дают возможность не читать, а быстро просматривать текст, обычно их выделяют жирным шрифтом. Выделенные слова должны представлять собой законченную мысль, поэтому часто необходимо изменить предложение так, чтобы оно содержало такую группу слов.

Чтобы не возникало визуального хаоса от разбросанных в тексте выделенных слов, ставьте опорные слова в начале блока текста.

Перечисления лучше оформлять списком. Такое оформление привлекает внимание, четко выделяет элементы, за счет этого текст удобно читать и легче понять.

Язык

Грамотность. Неграмотный текст придает информации недостоверный вид.

Сохраняйте тематическое единство текста и его соответствие заявленной в заголовке теме. Пишите о том, что вы обещали в заголовке. Нельзя позволять себе «растекаться мыслию по древу», вы рискуете потерять читателей еще до того, как они доберутся до понимания излагаемых вами идей.

Стилистическая направленность является ключевой для формирования вашей аудитории. Чем специфичнее язык, тем уже аудитория. В тоже время, использование в разумных пределах и с необходимыми пояснениями терминов, характерных для обсуждаемой темы, придаст большую авторитетность и убедительность тексту.

Стилистическое единство текста не менее важно, чем стилистическое единство дизайна страниц. Оно позволяет воспринимать ваш сайт (как источник информации), единым блоком, а не разрозненными кусками.

Исключением из этого правила являются статьи с других сайтов или написанные другими людьми на заказ. В этом случае надо явно указать, что текст написан другими людьми. Разнообразие — тоже хорошо.

Чувство меры.

Сохраняйте чувство меры, не возводите наши советы или советы других людей в абсолют. Помните, что лично вы отвечаете за успех вашего сайта.

**Делайте макеты страниц**

**Ч. 1**

Прежде чем приступить к HTML-верстке, сделайте макеты для каждого типа страниц. HTML-верстка очень кропотливое занятие и желательно иметь образец, по которому она будет делаться.

Инструмент для макетирования

Разрабатывать макет, пробовать варианты, оценивать внешний вид лучше в каком-нибудь WYSIWYG редакторе (What You See Is What You Get — что видишь, то и получишь). Я не имею ввиду WYSIWYG-редакторы для HTML, такие редакторы генерируют избыточный HTML код и не используются в работе настоящими веб-верстальщиками. Я имею в виду редактор, который позволит легко спроектировать внешний вид страницы: отформатировать текст, поменять цвет фона, вставить и масштабировать картинки, вставить таблицы. Такой редактор все хорошо знают — это Word.

Макет страницы

Макет страницы должен содержать все элементы, которые будут включены в страницы этого типа: логотип, рисунки, меню, образцы текста разного уровня (простой текст, заголовки, списки и т.п.).

Простой текст должен быть достаточной величины и содержать несколько абзацев, что бы можно было видеть величину отступов, межстрочных интервалов, пробелов, красных строк и других параметров оформления.

Списки тоже должны быть достаточной величины и содержать элементы, занимающие несколько строк.

### Черновая верстка

Сделайте в Word черновую верстку текста (одного текста, без логотипов, рисунков, меню) хотя бы нескольких страниц. Это позволит выбрать общий стиль верстки: шрифты, форматы заголовков, размеры отступов и т.д. Не надо ограничиваться одной страницей, т.к. весьма вероятно, что на одной странице встретятся не все варианты форматирования. Например, при HTML-верстке очередной страницы окажется, что нужен еще один уровень заголовков, а шрифт (или отступ) для него окажется слишком велик или, наоборот — мал.

Самый лучший вариант, это сделать черновую верстку всех страниц для которых достаточно материала. В конце этой работы ваш взгляд на то, как должен выглядеть текст на сайте может сильно измениться. Лучше, если это произойдет после черновой верстки, а не после HTML-верстки. Можно совместить черновую верстку с написанием текста страницы.

### Плюсы черновой верстки всех страниц

Сделав в Word черновую верстку текста всех страниц вы получите следующие преимущества:

Увидите все варианты форматирования

Word позволяет считать общее количество слов в документе.

Word позволяет считать количество вхождений указанного слова в документе. Это полезно для оптимизации текста под ключевые слова.

Word проверяет правописание.

При черновой верстке главным материалом, с которым приходится работать, является текст. О том, что текст должен легко читаться, быть понятным и хорошо структурированным мы уже говорили в статье Какой писать текст. В следующей статье мы немного расскажем о шрифтах.

## **Делайте макеты страниц**

Ч. 2

### **Шрифты**

Формы букв, способы их начертания совершенствовались писцами и типографами на протяжении веков. Любой шрифт — это результат творчества дизайнера и художника, по-этому, к шрифтам надо относиться с уважением.

Размер шрифта (кегель, font size) по традиции измеряется в пунктах. Один пункт равен 1/72 дюйма (1 дюйм = 2.54см.), или приблизительно, 0.35мм. Причем, высота определяется не по высоте заглавных букв, а с некоторым "припуском", так заглавные буквы шрифта в 10 пунктов имеют размер 7 пунктов.

Тип шрифта. Обычно используют шрифты двух основных типов: с засечками и без засечек.

Шрифты с засечками имеют небольшие дополнительные штрихи на концах основных штрихов. При этом нижние горизонтальные засечки образуют линию, вдоль которой как бы бежит глаз, что облегчает чтение больших объемов текста.

Пример: шрифт с засечками (шрифт Times New Roman). Roman в данном случае означает прямой шрифт.

Шрифты без засечек, или как их еще называют, рубленые. Типовое название таких шрифтов sans-serif (sans — без, serif — засечка). Первые шрифты этого типа были разработаны в начале XX века, их разработчики руководствовались идеями утилитаризма, проповедовавшего отказ от украшений, не имеющих реальной пользы.

Пример: шрифт без засечек (шрифт Arial).

В печатной типографике для заголовков обычно используют шрифты без засечек, а для основного текста шрифты с засечками, т.к. считается что их легче читать.

В Интернете шрифты с засечками не популярны, и весь текст обычно набирают шрифтами без засечек. Причина этого, видимо, кроется в низкой разрешающей способности экрана. В печатных текстах засечки имеют сложную, заостренную на конце, форму и соединяются с основным штрихом плавно. На экране отдельный символ занимает обычно 10–15 пикселей в высоту, засечка изображается линией толщиной всего в 1 пиксель и не может плавно соединяться с основным штрихом или иметь заостренную форму. Плюс к этому, для 10–15 пикселей символа 2 пикселя засечек (снизу и сверху) слишком большая визуальная нагрузка.

Немного особняком стоят моноширинные шрифты, имеющие постоянную ширину букв. Они хорошо подходят для набора текстов программ и денежных сумм в столбик. Т.е. там, где важно расположение символов в соответствующих позициях строго одного под другим.

Пример: моноширинный шрифт имитирует пишущую машинку (шрифт Courier New).

Отдельную группу составляют декоративные шрифты, имитирующие рукописные или национальные начертания. Их надо применять с большой осторожностью.

Некоторые советы по верстке текста

Минимум шрифтов. В большинстве случаев достаточно одного шрифта. Если одного шрифта мало, используйте шрифты с явно различными начертаниями. Присутствие в тексте слов, набранных похожими шрифтами будет раздражать читателя.

Используйте распространенные шрифты. Шрифта может не оказаться на компьютере пользователя. Некоторые шрифты появляются на компьютере во время установки программ, например, текстовых и графических редакторов. Таким образом, шрифт который есть у вас, не обязательно есть у всех пользователей. Есть возможность заставить браузер пользователя загрузить шрифт с web-сервера и использовать при просмотре страницы, но этот путь имеет 3 существенных недостатка:

Файлы шрифтов имеют приличный размер, порядка 100 К.

Установка шрифтов занимает время и может пройти не совсем гладко.

Вы можете нарушить авторский права на шрифт.

Не используйте курсивных (наклонных) вариантов шрифтов, например: это курсив (шрифт Arial + наклон). Причина та же — низкая разрешающая способность экрана и вызванная этим плохая "читабельность".

Пример: Курсивные шрифты с засечками — вообще на грани читабельности. Читать большие куски текста, набранного курсивом, сущая пытка.

Делайте шаблоны страниц.

После создания макетов страниц надо сделать HTML-шаблоны для всех типов страниц и CSS-файл стилей. Цель создания шаблонов — уменьшить количество ошибок и повторяющейся, рутинной работы во время верстки каждой отдельной страницы.

Требования к шаблону страницы

Шаблон должен содержать все элементы, которые будут включены в страницы этого типа: логотип, рисунки, меню, образцы текста разного уровня (простой текст, заголовки, списки и т.п.).

Простой текст должен быть достаточной величины и содержать несколько абзацев, что бы можно было видеть величину отступов, межстрочных интервалов, пробелов, красных строк и других параметров оформления.

Списки тоже должны быть достаточной величины и содержать элементы, занимающие несколько строк.

Если вы планируете ставить счетчики посещений, надо выделить и отметить на странице место для их размещения. Обычно код счетчика ставят вверху страницы, а логотип — внизу. Google не требует размещения своего логотипа, при размещении кода счетчика и подключении сайта к сервису Google Analytics.

Разделите структуру документа и визуальное оформление

Язык разметки гипертекстовых документов HTML ведет свою историю от языка SGML (1986 г.), разработанного для описания структуры документов. Предполагалось, что структура документа и визуальное оформление различных структурных единиц описываются в разных файлах и не зависят друг от друга. Т.е. мы можем взять файл с размеченным документом, поменять для него файл визуального оформления и получить документ, оформленный по-новому.

В первых версиях HTML этот принцип соблюдался и язык содержал в основном тэги структурной разметки. Однако, после того как был оценен коммерческий потенциал Интернета (точнее — его нового протокола http) и появились коммерческие веб-браузеры Netscape, а затем — Internet Explorer, их разработчики стали добавлять новые тэги визуальной разметки. В результате этого структурный и визуальный компоненты оказались перемешаны.

С появлением каскадных таблиц стилей (CSS) стало возможным разделить описание структуры документа и его визуальное оформление. Кроме того, что это более правильно с теоретической точки зрения, это еще и очень удобно. Для того чтобы изменить внешний вид многих страниц достаточно сделать исправления в одном месте — в CSS-файле стилей.

По этому старайтесь все визуальные инструкции (описания шрифтов, полей, отступов, цветов) вынести из HTML-страниц в файл CSS. Или, по крайней мере, собрать их в теге <style> </style> в заголовке (<head> </head>) HTML-страницы.

Еще одно важное преимущество CSS состоит в том, что они позволяют лучше управлять внешним видом документа. Например, тэг <p></p> добавляет строку пробелов до и после себя. Можно избавиться от этих пробелов, описав свойства тэга <p> в таблице стилей.

Считается, что HTML близок к исчерпанию своих возможностей и будет постепенно заменяться другими языками разметки. Один из таких языков это XML (тоже приложение SGML) Как простой переходный вариант сейчас широко используется язык XHTML, являющийся одновременно подмножеством XML и, при соблюдении простых правил, подмножеством HTML 4. Одно из таких правил состоит в том, что все тэги должны иметь закрывающую часть, например, тэг <br> должен записываться как <br />.

Суммируя все выше сказанное: делайте шаблоны страниц, используйте связку XHTML + CSS.

HTML-инструментарий.

Я не рекомендую WYSIWYG-редакторы для HTML, они не используются в работе настоящими веб-верстальщиками. По следующим причинам:

Генерируют избыточный HTML код.

HTML и CSS достаточно простые, их вполне можно освоить.

Если вы собираетесь делать сайты, изучив HTML и CSS вы многое поймете про сайты вообще.

Экстремальный вариант — верстать страницы в Notepad я тоже не рекомендую.

HTML-редактор должен иметь следующие возможности:

Быть удобным.

Иметь подсветку синтаксиса HTML и CSS кода.

Иметь встроенный браузер для просмотра HTML-страниц.

Желательно также наличие системы помощи по HTML и CSS.

Я рекомендую два HTML-редактора:

Web Dev Studio 2.0. Интерфейс русский, программа бесплатная. Удобный и симпатичный на вид, но система помощи слабовата. Очень удобно, что при закрытии запоминает список открытых файлов и при следующем запуске открывает их опять. Не смотря на то, что сайт проекта хостится на Народе (narod.ru), программа весьма качественная.

Вот какие возможности выделяет сам автор:

Поддержка: HTML, PHP, Perl, CSS, JS, VBS, ASP, SSI, cUrl, VRML, WML, XML

Полная справочная информация по названным технологиям

Проверка синтаксиса: HTML, XML, PHP, Perl

Настраиваемый сервис встроенного кода (Ctrl+Пробел)

Подсказка по параметрам функций (Shift+Ctrl+Пробел)

Сервис исследования и анализа кода

Интегрирование с буфером обмена, пользовательский буфер

Встроенный и внешний предварительный просмотр

Встроенный файловый менеджер

Быстрый поиск и замена по файлам

Многодокументальный интерфейс - MDI

Заготовки кода (Ctrl+J)

Сервис корректировки ввода

Сервис сравнения файлов

Переходы по элементам кода по Ctrl+Щелчок

Создание документа по шаблону, свои шаблоны

Интегрирование в систему, ассоциирование файлов с программой

Работа с макросами: создание, редактирование, воспроизведение

Расширенная подсветка синтаксиса, настройка подсветки

Множество настраиваемых установок

Настраиваемый интерфейс программы, различные стили

Работа с закладками в редакторе кода

Восстановление документов после сбоя в ОС

1st Page 2000 v2.00 Интерфейс английский, программа бесплатная. Тоже довольно удобный, но не такой симпатичный, имеет неудобное свойство — каждый раз надо открывать все файлы снова. Главный плюс — отличная система помощи по HTML и CSS, правда на английском. Основная часть файлов помощи располагается на компьютере пользователя, в особо скрупулезных случаях ссылка ведет на очень полезный сайт [www.htmlhelp.com](http://www.htmlhelp.com) с подробнейшей документацией по HTML и CSS, PR сайта 8(!) и ТИЦ 1000, такой высокий ТИЦ редкость для нерусскоязычного сайта.

Рекомендую работать в Web Dev Studio, а в 1st Page 2000 использовать справочную систему.

Основной курс

Имя домена, whois, опасность ухода имен

Прежде чем опубликовать свой сайт в Интернете необходимо придумать и зарегистрировать для него домен. Слово «домен» происходит от английского «domain» — владение, имение, территория, область, сфера. Каждый домен в Интернете определяется своим именем. Обычно вместо имя домена говорят просто домен.

Правила записи доменных имен.

Допустимые символы:

Можно использовать буквы, цифры и «-» (дефис, минус), регистр символов значения не имеет.

Минимальная длина имени 2 символа, максимальная — 255 символов (см. уточнение ниже).

Имя не может начинаться или заканчиваться на дефисы или содержать два дефиса подряд.

В именах доменов обычно используются латинские буквы. Можно использовать и кириллицу, но ее использование имеет ряд недостатков, например, что бы набрать www.МойДомен.ru надо дважды переключить язык, плюс разночтения с буквами похожими по начертанию в латинице и кириллице: а, с, о, р, у.

Имя домена состоит из нескольких частей, разделенных точкой:

Имя домена первого уровня — из одной части (com, ru, org, biz).

Имя домена второго уровня — из двух частей (google.com, yandex.ru).

Имя домена третьего уровня — из трех частей (news.google.com, wordstat.yandex.ru).

Длина каждой части не более 63 символов.

Общая длина имени домена с точками и именем домена первого уровня — 255 символов.

Доменные имена или, как говорят, пространство доменных имен, построено иерархически, т.е. google.com поддомен домена com, news.google.com — поддомен домена google.com. Домены первого уровня еще называют доменными зонами, например: google.com — домент в зоне com.

Домен надо регистрировать.

Имя домена должно быть уникальным. Поэтому имена доменов первого и второго уровней регистрируют специальные организации. Имена доменов третьего уровня выдают владельцы доменов второго уровня. Например, если вы владеете доменом MyDomain.com, вы можете создать поддомены SubDomain1.MyDomain.com, SubDomain2.MyDomain.com и т.д.

Домены первого уровня не регистрируются на частных лиц. Они являются общими для всего Интернета и служат для объединения сайтов по типам. Например:

COM — коммерческие сайты.

ORG — сайты некоммерческих организаций.

RU — русскоязычные.

US — сайты из USA.

Частным лицам и организациям на определенный срок выдаются, или как говорят — делегируются (фактически, сдаются в аренду), домены второго уровня. Стоимость регистрации домена на 1 год составляет порядка 10–15 долларов в зоне COM и 20 долларов в зоне RU. Иными словами, вы выбираете доменную зону, придумываете имя, которого еще нет в выбранной зоне, платите деньги и закрепляете это имя за собой для своего сайта на определенный срок.

С www или без?

Имя домена не включает в себя префикс WWW. WWW — это имя службы или сервиса, в данном случае — службы world wide web — предоставления информации в виде гипертекста по протоколу HTTP. Т.е., www.site.com говорит о том, что по этому адресу site.com предоставляет сервис WWW, а не другие сервисы, например: FTP.

Выбор доменного имени.

Несколько простых правил для доменного имени:

По возможности короткое.

Хорошо запоминается.

Легко и однозначно воспринимается на слух, например по телефону.

Имет смысл — совпадает или ассоциируется с названием организации или сферой деятельности.

Желательно, чтобы домен в смежной зоне был свободен. Для Рунета смежные зоны это RU — COM. Если человек набирает имя домена по памяти, он может ошибиться и набрать вместо RU — COM и наоборот. Желательно иметь право на оба домена, тогда, в любом случае, посетитель для вас не пропадет.

Если название вашей фирмы имеет простой и однозначный вариант латинского написания, можно использовать его.

В связи с тем, что уже зарегистрировано огромное количество доменов (около 700 000 в зоне RU), подобрать хорошее имя, удовлетворяющее всем приведенным правилам, очень сложно.

Облегчить задачу подбора имени домена могут следующие соображения:

Если вы не планируете рекламировать ваш сайт в off-line, печатать его название на визитках, диктовать по телефону и т.д., то длина названия и запоминаемость не играют особой роли. Адрес сайта легко запомнить в избранном или как ярлык на рабочем столе. Но в этом случае очень важно содержимое тега <Title>, особенно первые 2–3 слова (см. Оптимизация сайта для поисковых систем), так как содержимое <Title> отображается как название ярлыка или элемента в избранном.

Поисковые системы учитывают наличие слов из запроса в имени домена (и в именах файлов тоже) и считают документы с таких сайтов более релевантными. Если невозможно подобрать название из одного слова — используйте два или три главных слова из семантического ядра. Рекомендуется разделять слова знаком «-» (минус). Не рекомендуется в таких случаях использовать транслит (запись русских слов латинскими буквами) .

Можно, например, воспользоваться сервисом подбора имен на webnames.ru. Он проверяет доменные имена, составленные их комбинации двух или трех английских ключевых слов.

Как проверить, занят домен или свободен

Для проверки занятости доменов существуют специальные сервисы Whois. В Интернете огромное количество сайтов предоставляет услуги Whois-сервиса. Они работают следующим образом:

Вы задаете имя домена второго уровня.

Если домен свободен — сервис отвечает, что такого домена нет или он не занят или еще что-нибудь в таком роде.

Если домен занят — сервис сообщает дату регистрации домена, дату окончания срока действия регистрации и другую дополнительную информацию: имя владельца, записи DNS хостинг-провайдера.

Увод имен доменов

Учитывая, что подобрать хорошее имя трудно, имена тоже стали товаром. Некоторые люди (кибер-сквоттеры) регистрируют хорошие имена и потом продают их, т.е. предлагают переоформить регистрацию на имя покупателя за деньги. Откуда же они берут эти хорошие имена? Некоторые придумывают сами, а некоторые — воруют. Мне приходилось читать на форумах про случаи «увода» имен после того, как их проверяли через Whois сервисы. Вывод такой: не надо проверять домен на занятость через Whois сервис на неизвестном вам сайте.

Для начала вполне достаточно Whois-сервиса Центра регистрации доменов в зоне RU. Он предоставляет информацию о доменах в зонах RU, SU, COM, NET, ORG, BIZ, INFO и об IP-адресах.

Если имя выбрано, можно зарегистрировать домен. Можно отложить регистрацию домена до того момента, как будет готов сайт и решен вопрос с хостингом, но если вы нашли удачное имя и есть вероятность, что его могут занять — не надо тянуть, надо регистрировать.

## Регистрация домена

Есть два способа регистрации доменного имени — самостоятельно и через хостинг-провайдера. Сейчас хостинг-провайдеры предлагают зарегистрировать домен через них или, даже, предлагают домен бесплатно при оплате некоторых тарифных планов. В любом случае надо знать процедуру и ключевые моменты регистрации доменного имени.

Что является ключевым при регистрации домена.

При регистрации домена самое важное — становитесь ли вы хозяином этого домена. Хозяин домена тот, кто может сделать все следующее:

Войти в учетную запись домена.

Изменить пароль на вход в учетную запись.

Изменить регистрационные данные и e-mail указанный для связи с вами.

Изменить записи указывающие на DNS вашего хостинг-провайдера. Изменив эти записи так, чтобы они указывали на другого провайдера, вы смените провайдера. DNS хостинг-провайдера (хостера) можно, узнать у него на сайте.

Оплатить продление регистрации домена.

Если вы решили зарегистрировать домен сами.

Зарегистрировать домен самостоятельно довольно просто. Есть сервисы, предлагающие услуги регистрации доменов, плюс к этому, все хостинг-провайдеры наряду с хостингом предлагают услуги по регистрации доменов. В этом случае все регистрационные формы вы заполняете сами, платите электронными деньгами или карточкой или банковским переводом и владеете всей ключевой информацией о вашем домене.

Система регистрации доменов.

Для полноты картины кратко опишем систему регистрации доменов. Для того, чтобы зарегистрировать домен, знать как устроена система регистрации необязательно, но полезно.

ICANN (Internet Corporation for Assigned Names and Numbers — корпорация Интернета для специализированных адресов и номеров) осуществляет управления системой записи доменных зон, или как их еще называют — TLD (Top Level Domain — домены верхнего уровня, домены первого уровня). Различают TLD двух видов:

g TLD (generic Top Level Domain) — общие домены первого уровня, такие как COM, NET, BIZ и т.п.

cc TLD (country code Top Level Domain) — национальные домены первого уровня (домены стран), такие как RU, US и т.п.

Проще говоря, ICANN придумывает и ведет список доменов верхнего уровня.

Операторы доменов верхнего уровня. Или, как их еще называют, администраторы реестра. Это организации, которые ведут базу данных (реестр) доменов второго уровня на определенном TLD и предоставляют интерактивный доступ к ней для регистрации доменов. Они же обеспечивают сервис Whois по своему домену верхнего уровня. В связи с тем, что для этого надо содержать персонал, технику, покупать программное обеспечение, операторы TLD берут деньги за регистрацию доменов. Стоимость доменов в зоне RU около \$20, стоимость доменов в зоне COM порядка \$10–15 для конечного пользователя.

Регистраторы. Организации, которые предоставляют услуги по регистрации доменов второго уровня, прием платежей, техническую поддержку пользователей и т.п. Они должны заключить договор с соответствующими операторам TLD и получить аккредитацию (одобрение) ICANN, которая проверяет техническую готовность и подготовку персонала.

Партнеры, реселлеры. Организации, или частные лица которые предоставляют услуги по регистрации доменов, «покупая оптом» у регистраторов и «продавая в розницу» пользователям.

Процесс регистрации домена.

Выбрать сайт, предоставляющий услуги регистрации доменов.

Он должен иметь удобный для вас способ оплаты.

Хорошо бы найти людей, регистрировавших через него сайты, и узнать их впечатление. Можно спросить совета на форумах.

Лучше, если это будет сайт регистратора. Хотя иногда бывает, что реселлеры, работающие с иностранными регистраторами, предлагают лучшие цены, чем российские регистраторы, для зон COM, NET и подобных.

Завести учетную запись.

Подать заявку на регистрацию и внимательно заполнить предлагаемые формы, особенно контактный e-mail и DNS хостера. Все введенные данные можно потом изменить, но лучше не ошибаться.

Оплатить регистрацию. Обычно, деньги можно положить заранее и использовать их по мере надобности.

Домен будет готов к работе в течение от нескольких часов до одних суток. После регистрации домена регистратор (или реселлер) должны прислать вам письмо с подтверждением.

После регистрации домена.

Надо помнить на какой срок вы зарегистрировали домен и не забывать продлевать регистрацию. Перед истечением срока регистрации домена регистратор должен прислать по e-mail письмо с напоминанием, но лучше помнить самому.

## Выбор хостинга

Для того, чтобы ваш сайт был доступен в Интернете его надо разместить на web-сервере, подключенном к Интернету. Есть два варианта:

Разместить web-сервер на одном из имеющихся в наличии компьютеров и подключить его (например, по выделенной линии) к интернет-провайдеру. Так иногда поступают организации. Этот вариант требует наличия грамотных IT-специалистов и быстрого канала, по которому производится подключение.

Второй, более распространенный вариант — разместить сайт на сервере организации, профессионально занимающейся хостингом — размещением сайтов клиентов на своих серверах и предоставлением всех необходимых сервисов. В этом случае вам не нужно заботиться о техническом обслуживании оборудования и настройке программ. Всю ответственность за обеспечение стабильной работы вашего сайта возьмут на себя специалисты хостинг-провайдера.

Мы будем рассматривать второй вариант — размещение сайта у хостинг-провайдера (хостера). Выбор хостинга — дело ответственное и на первых порах сложное. В этой статье я сообщу некоторую общую информацию, дам ссылку на очень полезный цикл статей по выбору хостинга и поделюсь своим опытом.

Какие бывают хостеры.

Хостеры бывают настоящие, которые имеют свое оборудование, персонал, лицензию и виртуальные, которые как партнеры-реселлеры крупных хостеров перепродают услугу и оказывают некоторую техническую поддержку.

Где может физически располагаться хостинг.

Сами сервера могут располагаться где угодно, например, хостер может иметь оборудование в Москве и арендовать оборудование в США.

Желательно, чтобы сервера располагались на технической площадке, имеющей хорошие каналы связи. Одна из самых известных в Рунете площадок — точка обмена трафиком M9, созданная первыми российскими интернет-провайдерами, чтобы трафик между серверами российского сегмента сети ходил не через океан и заграничные каналы, а внутри России.

Какой бывает хостинг.

В зависимости от потребностей хостинг может быть разного уровня:

Виртуальный сервер. Вам выделяется место на одном из серверов, вместе с вашим сайтом там будут располагаться и другие сайты. Сервер администрирует персонал хостера, на сервере действуют общие для всех настройки. Вы управляете своим сайтом через панель управления, обычно это DirectAdmin или cPanel.

Физический выделенный сервер. Вам предоставляется отдельный сервер. Вы можете сами удаленно администрировать его или оплачивать услуги администрирования персонала хостера. Вы можете делать свои настройки.

Виртуальный выделенный сервер. Сервер эмулируется программно, но его можно отдельно администрировать и делать свои настройки.

Размещение своего сервера на площадке хостинг-провайдера.

Какие бывают цены.

Для небольших сайтов можно найти хороший хостинг от \$1 в месяц.

Как выбирать хостинг.

Лучше всего, конечно, посоветоваться со знакомыми или посоветоваться на форумах. Для начала рекомендую прочитать цикл статей Как выбрать провайдера? на сайте hostobzor.ru.

Какого хостера могу посоветовать я сам.

Могу посоветовать RuWeb.net, сервера в Москве и США, цены на хостинг от \$1, один месяц бесплатного тестового хостинга. Сразу включено:

Место на диске: 10 Мб.

Исходящий трафик: 500 Мб/мес.

FTP аккаунты: 99

POP3 ящики: 99

Поддомены: 99

Листы рассылки: 99

Бесплатно предоставляется 1 mySQL база на каждые 10 Мб места (если заказана поддержка mySQL).

При оплате за 6 месяцев и более: скидка 5%.

При оплате за 12 месяцев и более: скидка 10%.

Минимальная цена тарифного плана: 27 руб./мес (для юр.лиц: 80 руб./мес). Минимальная сумма к оплате: 100 руб.

Очень удобно, что отсутствуют фиксированные тарифные планы, можно выбирать необходимые опции и ресурсы сугубо индивидуально, цена рассчитывается соответственно выбранному.

Все процессы полностью автоматизированы, в том числе оплата. Саппорт быстрый, есть форум на котором можно задавать вопросы с большим количеством уже отвеченных вопросов.

Как я нашел RuWeb.net: я зашел на Яндекс.Деньги, далее на страницу хостеров, которые принимают Яндекс-Деньги, походил по ссылкам, сравнил цены и условия. У RuWeb.net мне понравились условия и то, что я их нашел всерьезном месте — на сайте Яндекса.

Еще один плюс RuWeb.net. Когда я размещал свой первый сайт, тот хостер у которого я размещал сайт, дал мне ссылку на руководство по работе с панелью управления сайтами DirectAdmin, это одна из двух самых популярных панелей (вторая cPanel). Сайт очень полезный и добротно сделанный. Так вот, этот сайт сделан Ruweb.net. Спустя некоторое время, когда я искал хостинг и остановился не Ruweb, я вспомнил про тот сайт и это было еще одним доводом в их пользу.

Короче говоря, рекомендую RuWeb.net, особенно для начала. Зайдите, думаю вам понравится:

RuWeb.net

Хостинг в Москве и США по индивидуальным тарифам,  
регистрация доменов, SSL-сертификаты.

Хостинг для реселлеров и аренда выделенных серверов в Москве.

## Продвижение сайта

Сайты создаются для того, что бы их посещали. Деятельность по привлечению посетителей называется продвижением. Самый первый шаг в продвижении — регистрация сайта в поисковых системах. Самый простой способ продвижения, самый надежный и дающий наиболее долговременный эффект — создание сайта с большим количеством хороших, интересных, грамотно оптимизированных текстов.

Продвижение с помощью текстов.

Продвижение сайта путем наполнения его хорошим текстовым содержанием имеет ряд неоспоримых преимуществ.

Хороший текст — это законсервированная информация, а люди ищут в Интернете именно информацию. Посетитель, пришедший на сайт с большим количеством интересных и полезных текстов, занесет его в избранное, или сделает ярлык на рабочем столе, будет приходить снова и снова. Даже, если человек ищет, например, музыку, он ищет ее по имени исполнителя, по названию песни, т.е. сначала он ищет текст.

Много текста — это большое количество словосочетаний, многие из которых могут совпадать с запросами пользователей в поисковых системах. Поисковые системы — источник большой доли целевых посетителей вашего сайта.

Текст не исчезнет, в отличие от внешних ссылок. Даже, если ссылка не куплена, а поставлена по инициативе владельца другого сайта, она может быть убрана, сайт может перестать существовать и т.п. Если же ссылка куплена - она, плюс к тому, требует регулярной оплаты. Текст, написанный один раз и размещенный на сайте, будет работать день за днем на посещаемость вашего сайта, пока вы его сами не удалите.

Хорошие тексты могут приносить внешние ссылки. Если вашу статью перепечатают на другом сайте, то должны поставить на вас ссылку.

Внешние ссылки.

Современные поисковые системы наряду с содержанием страниц учитывают и индекс цитируемости страницы или всего сайта, то есть, ссылки с других сайтов на ваш сайт (внешние ссылки). Поисковые системы считают количество и качество внешних ссылок мерилom авторитетности сайта. При этом учитываются:

тема ссылающейся страницы;

текст ссылок;

количество ссылок на ссылающейся странице.

Способы увеличения количества внешних ссылок.

Существует несколько способов добывания внешних ссылок:

Улучшать сайт, его наполнение, полезность и ждать, что другие web-мастера поставят ссылки на ваш сайт, как на источник полезной информации.

Размещать свои статьи со ссылкой на других ресурсах.

Размещать в каталогах ссылки на сайт.

Общаться на форумах, оставлять свои комментарии с указанием ссылки.

Обмениваться ссылками с другими сайтами.

Покупать ссылки.

Легко догадаться, что поисковые системы приветствуют не все способы получения ссылок. Первые два способа законные. Третий — частично, есть авторитетные каталоги, есть — не очень, есть и такие, которые удаляют из индекса (банят). Ссылки с форумов не учитывает Яндекс. Обменные и покупные ссылки отсеиваются по мере развития алгоритмов их обнаружения. Но пока с разной степенью надежности работают все способы.

Реклама в Интернете.

Реклама на других сайтах. Вы покупаете у другого web-мастера право разместить свою рекламу на его сайте.

Контекстная реклама. Вы платите службе контекстной рекламы за показ вашей рекламы на сайтах подходящих по своей тематике. Например, на сайтах про ремонт квартир покажут рекламу красок и растворителей.

Почтовая рассылка.

Можно бесплатно завести свою рассылку на серверах почтовых рассылок [Subscribe.ru](http://Subscribe.ru), [MailList.ru](http://MailList.ru), [Content.mail.ru](http://Content.mail.ru). Вести такую рассылку надо постоянно, регулярно уделяя время на подбор материала и ответы на вопросы читателей.

Ваши статьи в чужих рассылках.

Если рассылка выходит регулярно, а в ином случае нет смысла связываться, автор рассылки постоянно ищет для нее материал. Предложите свою статью со ссылкой на ваш сайт. При этом статья не должна иметь откровенно рекламного характера. Списки рассылок, количество подписчиков и периодичность можно посмотреть там же — на [Subscribe.ru](http://Subscribe.ru), [MailList.ru](http://MailList.ru), [Content.mail.ru](http://Content.mail.ru).

Общение на тематических форумах.

Мы уже писали про форумы, в этом случае имеется ввиду привлечение посетителей с форума на сайт. Например, вы пишете на форуме интересный пост и в подписи адрес сайта, многим будет интересно посмотреть на сайт такого умного человека.

Подпись в письмах.

Включите адрес и краткое (в одну фразу) описания сайта в подпись в ваших e-mail.

Рекламные и печатные материалы.

Включайте адрес сайта в рекламу вашей фирмы, визитные карточки, буклеты, бланки, прайсы, письма и т.п.

# Perl

---

**Perl** — высокоуровневый интерпретируемый динамический [язык программирования](#) общего назначения, созданный [Ларри Уоллом](#), лингвистом по образованию. Название языка представляет собой [аббревиатуру](#), которая расшифровывается как *Practical Extraction and Report Language* — «практический язык для извлечения данных и составления отчётов»<sup>[1]</sup>. Первоначально аббревиатура состояла из пяти символов и в таком виде в точности совпадала с английским словом *pearl* («жемчужина»). Но затем стало известно, что такой язык существует (см. [PEARL](#) ([англ.](#))), и букву «a» убрали. Талисманом языка Perl является [верблюд](#) — не слишком красивое, но очень выносливое животное, способное выполнять тяжёлую работу.

Основной особенностью языка считаются его богатые возможности для работы с текстом, в том числе работа с [регулярными выражениями](#), встроенная в синтаксис. Перл унаследовал много свойств от языков [Си](#), [AWK](#), скриптовых языков [командных оболочек UNIX](#).

Perl также знаменит огромной коллекцией [дополнительных модулей CPAN](#), находящейся по адресу <http://www.cpan.org>.

## История

---

### [\[править\]](#) Первые версии

[Ларри Уолл](#) начал разработку Perl в 1987 году, когда работал программистом в [Unisys](#)<sup>[2]</sup>. Версия 1.0 была выпущена и анонсирована в [новостной группе](#) comp.sources.misc 18 декабря 1987 года<sup>[3]</sup> как «замена» для [awk](#) и [sed](#).

Perl 2 был выпущен в 1988 году. Основным нововведением в нём был переработанный механизм [регулярных выражений](#). Perl 3, выпущенный в 1989 году, получил возможность обрабатывать потоки [двоичных данных](#).

Изначально, единственной документацией для Perl была единственная (огромной длины) [man](#)-страница. В 1991 году была выпущена книга «*Programming Perl*» (известная многим программистам как «Верблюжья книга» (*Camel Book*) из-за изображения на обложке), которая, [де-факто](#), стала стандартом языка. В то же самое время, версия языка была «поднята» до 4 — не столько, чтобы указать на значительные изменения, сколько, чтобы обозначить, что эта версия языка документирована книгой.

### [\[править\]](#) Ранний Perl 5

Perl 4 прошёл через серию релизов, остановившись на Perl 4.036 в 1993 году. В этот момент, Ларри Уолл забросил Perl 4 и начал работу над Perl 5. Разработка Perl 5 была начата в 1994 году. В то же время был создан [список рассылки perl5-porters](#) для координации работы над [портированием](#) Perl 5 на различные платформы. Он до сих пор остаётся основным форумом по разработке, обслуживанию и портированию Perl 5<sup>[4]</sup>.

Perl 5.000 был выпущен 17 октября 1994 года.<sup>[5]</sup> Он включал в себя полностью переписанный [интерпретатор](#), а также много новых языковых возможностей, таких,

как [объекты](#), [ссылки](#), [локальные переменные](#) (`my $var_name`) и [модули](#). Особенно важной частью были модули, так как они предоставили механизм расширения языка без модифицирования интерпретатора. Это позволило стабилизировать интерпретатор, но при этом, дало возможность обычным Perl-программистам добавлять в язык новые возможности. Perl 5 находится в активной разработке по сей день.

Perl 5.001 был выпущен 13 марта 1995 года. Perl 5.002 увидел свет 29 февраля 1996 года с новой поддержкой прототипов. Это позволило авторам модулей создавать функции, которые вели себя так же, как и встроенные функции Perl. Версия Perl 5.003 появилась 25 июня 1996 года и устранила обнаружившиеся проблемы с безопасностью.

Одно из важнейших событий в истории Perl 5 случилось за пределами собственно языка и было следствием поддержки модулей. 26 октября 1995 года начала свою работу [CPAN](#) (*Comprehensive Perl Archive Network* — «всеобъемлющая сеть архивов Perl»), которая стала [репозиторием модулей Perl](#), а также исходного кода самого языка. На данный момент сеть насчитывает более 20 000 модулей, созданных более чем 8 000 программистами<sup>[6]</sup>.

Perl 5.004 был выпущен 15 мая 1997 года и содержал, кроме всего прочего, пакет UNIVERSAL, который давал языку основной объект, от которого автоматически происходили все классы. Была также включена возможность запрашивать версию модулей. В дополнение к этому Perl стал поддерживать Microsoft Windows, а также ряд других операционных систем.

Perl 5.005 увидел свет 22 июня 1998 года. Этот релиз включал в себя несколько улучшений процессора регулярных выражений, новые перехваты в бэкэнд с помощью модулей `B : *`, оператор кавычек `qx//` для регулярных выражений, большой выбор других новых модулей, а также поддержку ряда операционных систем, в частности, [BeOS](#).

## [\[править\]](#) **С 2000 по наше время**

Perl 5.6 был выпущен 22 марта 2000 года. Главные изменения включали поддержку 64-битных систем, представление строк в стандарте [Юникод](#), поддержку больших файлов (файлы больше 2 Гб) и ключевое слово «our»<sup>[7][8]</sup>. Во время разработки Perl 5.6 было принято решение изменить схему наименования версий на более близкую к другим проектам с открытым исходным кодом. За версией 5.005\_63 следовала версия 5.5.640; согласно плану, версиям в разработке должны были присваиваться нечётные номера, стабильным версиям — чётные.

В 2000 году Ларри Уолл выступил с призывом к сообществу активно обсуждать предложения к следующей версии Perl. Результатом этого обсуждения стал 361 документ [RFC](#) (*Request for Comments* — «запрос комментариев»), которые были использованы в разработке [Perl 6](#). В 2001 году<sup>[9]</sup> была начата работа над финальным документом, подводящим итог предложениям к созданию нового поколения Perl. Результат был представлен не в виде формального документа, а в качестве дайджеста всех RFC. К этому времени Perl 6 существовал исключительно в виде описания языка.

Perl 5.8 был впервые выпущен 18 июля 2002 года и с этого момента получал ежегодное обновление. Последняя версия Perl 5.8 — 5.8.9 увидела свет 14 декабря 2008 года. Perl 5.8 улучшал поддержку

Юникода, добавлял новую реализацию ввода/вывода, добавлял поддержку многопоточности, увеличивал числовую точность и добавлял несколько новых модулей<sup>[10]</sup>.

В 2004 году началась работа над Синописом (Synopsis) — первоначально задумавшимся как ряд документов, подводящих итоги финального документа, упоминавшегося выше. Однако постепенно этот ряд документов и стал спецификацией Perl 6. В феврале 2005 года Audrey Tang начал работу над Pugs<sup>[11]</sup> — интерпретатором [Perl 6](#), написанном на [Haskell](#)<sup>[12]</sup>. Это было первой настоящей попыткой воплотить Perl 6 в реальность. Эта попытка заглохла в 2006 году.

18 декабря 2007 года, в день 20-летней годовщины Perl 1.0, была выпущена версия Perl 5.10.0. Эта версия содержала в себе ряд заметных нововведений, приближающих её к [Perl 6](#). Одними из таких нововведений стали новые операторы [switch](#) (названных «given»/«when»), обновление регулярных выражений, а также «умный» оператор совпадения «[~~](#)»<sup>[13][14]</sup>.

Примерно в это же время всерьёз началась разработка новой реализации Perl 6, известной как Rakudo Perl<sup>[15]</sup>, разработанной в тандеме с виртуальной машиной [Parrot](#). С ноября 2009 года Rakudo Perl регулярно обновляется каждый месяц и является на данный момент самой полной реализацией [Perl 6](#).

Значительное изменение процесса разработки Perl 5 произошло после появления Perl 5.11.

Сообщество разработчиков перешло на ежемесячный цикл релизов, с планированием даты релиза на три месяца вперёд.

12 апреля 2010 года Perl 5.12.0 был представлен публике. Выдающиеся нововведения включают в себя поддержку нового синтаксиса `package NAME VERSION`, оператор `Yada Yada` (служащего для маркировки кода-заполнителя, который ещё не реализован), полное решение [Проблемы 2038 года](#) (Y2038), перегрузку операторов регулярных выражений, поддержку [DTrace](#) (фреймворка динамической трассировки), а также полную поддержку стандарта Unicode 5.2<sup>[16]</sup>. 7 сентября 2010 года был выпущен Perl 5.12.2. Этот релиз содержит обновление модулей, а также некоторые изменения в документации<sup>[17]</sup>.

## [\[править\]](#) Perl 6

Основная статья: [Perl 6](#)

С 2000 года идёт разработка новой (6-й) версии языка. В отличие от предыдущих версий, разработчики планируют создать чётко определённый стандарт языка. В настоящее время существуют экспериментальные компиляторы [Perl 6](#), но продолжается дальнейшая дискуссия о новых правилах. На русском языке информация о новой версии языка доступна на сайте [perl6.ru](#).

## [\[править\]](#) Обзор

---

Perl — язык программирования общего назначения, который был первоначально создан для манипуляций с текстом, но на данный момент используется для выполнения широкого спектра задач, включая [системное администрирование](#), [веб-разработку](#), [сетевое программирование](#), игры, [биоинформатику](#), разработку [графических пользовательских интерфейсов](#).

Язык можно охарактеризовать скорее как практичный (лёгкость в использовании, эффективность, полнота), чем красивый (элегантность, минималистичность)<sup>[18]</sup>. Главными достоинствами языка являются поддержка различных [парадигм](#) ([процедурный](#), [объектно-ориентированный](#) и [функциональный](#) стили программирования), контроль за памятью (без сборщика мусора, основанного на циклах), встроенная поддержка обработки текста, а также большая коллекция [модулей](#) сторонних разработчиков.

Согласно [Ларри Уоллу](#), у Perl есть два девиза. Первый — «Есть больше одного способа это сделать» («*There's more than one way to do it*»), известный также под аббревиатурой [TMTOWTDI](#). Второй слоган — «Простые вещи должны оставаться простыми, а сложные — стать выполнимыми» («*Easy things should be easy and hard things should be possible*»).

## [\[править\]](#) Особенности

Общая структура Perl в общих чертах ведёт своё начало от языка [Си](#). Perl — [процедурный](#) по своей природе, имеет [переменные](#), выражения [присваивания](#), блоки кода, отделяемые [фигурными скобками](#), управляющие структуры и [функции](#).

Perl также заимствует ряд свойств из языков программирования [командных оболочек UNIX](#). Все переменные маркируются ведущими знаками, которые точно выражают тип данных переменной в этом контексте (например, скаляр, массив, хеш). Важно, что эти знаки позволяют переменным быть интерполированным в строках. Perl обладает множеством встроенных функций, которые обеспечивают инструментарий, часто используемый для программирования оболочки, например сортировку или вызов системных служб.

Perl заимствует [массивы](#) из [Lisp](#), [регулярные выражения](#) из [AWK](#) и [sed](#), из AWK также позаимствованы [хешы](#) («ассоциативные массивы»). Регулярные выражения облегчают выполнение многих задач по парсингу, обработке текста и манипуляций с данными.

Perl 5 добавил поддержку сложных типов данных, [первоклассных функций](#) ([замыкание](#) как значение) и объектную модель. В последнюю входят [ссылки](#), пакеты, выполнение методов от класса, переменные с лексическим объявлением [области видимости](#), а также [директивы](#) компилятору (например, `strict`). Главнейшим усовершенствованием, представленным в Perl 5, стала возможность помещать код в «пакеты» (*package*) в качестве модулей для повторного использования. Ларри Уолл позже заметил, что «Весь замысел модульной системы Perl 5 сводился к поощрению роста культуры Perl, а не строчек кода»<sup>[19]</sup>.

Все версии Perl выполняют автоматическую типизацию данных и автоматический контроль над памятью. Интерпретатор знает тип и запросы памяти каждого объекта программы, он распределяет и освобождает память, производя [подсчёт ссылок](#). Перевод одного типа данных в другой — например, числа в строку — происходит автоматически во время исполнения, невозможные для выполнения переводы типов данных приводят к фатальной ошибке.

## [\[править\]](#) Синтаксис языка Perl

Синтаксис Perl имеет много общего с синтаксисом языков [Си](#), [AWK](#), [sed](#) и [Bourne shell](#).

Первая строка исходного кода может начинаться с «#!/Путь/к/Perl [-ключи]» — что указывает системе путь к интерпретатору Perl для выполнения программы в UNIX системах и выполнения их на [веб-сервере](#). Изначально обработка такой строки — это функция [shell](#), но Perl при выполнении программы также проверяет эту строку и читает из неё ключи, игнорируя путь к интерпретатору.

[\[править\]](#)**Пример программы**

Простейшая программа «[Hello, world!](#)» выглядит следующим образом:

```
#!/usr/bin/perl
print "Hello, world!\n";
```

либо для [CGI](#):

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "Hello, world!";
```

либо используя один из post modern web-framework

```
use Mojolicious::Lite;
get '/' => {text => 'Hello World!'};
app->start;
```

либо используя [PSGI](#) спецификацию

```
my $app = sub {
    return [200, ['Content-Type' => 'text/plain'], ["hello, world\n"]];
}
```

Дозапись в строку.

```
$x = 5;
$x .= 0;
print $x; #50
```

А вот так можно напечатать число [гугол](#).

```
print 1, 0 x 100;
```

[\[править\]](#)**Типы данных**

Основные типы данных: [скаляр](#), [массив](#), [хеш-таблица](#), [функция](#), [указание на файл](#), [запись таблицы](#) [символов](#). Переменные разных типов отличаются знаком, который стоит перед именем переменной:

| Тип                    | Символ | Пример | Описание                                                                                                              |
|------------------------|--------|--------|-----------------------------------------------------------------------------------------------------------------------|
| <a href="#">Скаляр</a> | \$     | \$foo  | единичное значение; может быть числом, <a href="#">строкой</a> , указанием на файл, а также <a href="#">ссылкой</a> . |
| <a href="#">Массив</a> | @      | @foo   | Упорядоченная коллекция скаляров.                                                                                     |

|                                         |     |               |                                                                                                                                                     |
|-----------------------------------------|-----|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Хеш-таблица</a>             | %   | %foo          | Коллекция из пар строка-скаляр; строки называются <i>ключами</i> , а скаляры — <i>значениями</i> . Синоним — <a href="#">ассоциативный массив</a> . |
| <a href="#">Файловый дескриптор</a>     | < > | \$foo или FOO | Представление открытого файла или другой цели, открытой для чтения и/или записи.                                                                    |
| <a href="#">Функция</a>                 | &   | &foo          | Отрезок кода, который принимает аргументы, выполняет действия и возвращает значения.                                                                |
| <a href="#">Запись таблицы символов</a> | *   | *foo          | Запись таблицы символов для всех типов с именем 'foo'.                                                                                              |

## [\[править\]](#) Скаляр

- Скалярные переменные используются для хранения одиночных значений. Они могут содержать числа, строки и [ссылки](#) на другие [объекты](#). Перед именем скалярной переменной необходимо ставить знак доллара '\$'. Тип скалярной переменной не фиксирован (в отличие от, например, языка [Си](#)) и определяется динамически в зависимости от контекста. Например,

```
$x = 10; # число
$y = $x + 1; # используем $x как число...
$z = $x . 'string'; # ... а теперь — как строку
$ref = \ $x; # $ref является указателем на $x
$$ref = 10; # $x содержит значение 10
```

Строковые данные должны быть помещены в кавычки. Perl использует два вида кавычек — одиночные и двойные. Помещение строки в двойные кавычки позволяет именам переменных, содержащимся в строке, автоматически заменить собою название переменных. Заключение строк в одиночные кавычки предотвращает такое поведение:

```
## Пример интерполяции переменных в двойных кавычках

$pi          = 3.141592654;
$var1        = 'Значение Pi - $pi\n';
print $var1;                                     # Значение Pi - $pi\n
$var2        = "Значение Pi - $pi\n";
print $var2;                                     # Значение Pi -
3.141592654
```

Для включения двойных кавычек в строку, которая была помещена в двойные кавычки, следует использовать отмену знака — \". Для включения одинарных кавычек в строку из одинарных кавычек — \'. Для удобства работы со строками, содержащими много знаков кавычек внутри, Perl позволяет использовать альтернативные способы закавычивания. Альтернативная форма использует любую пару знаков, помещённых после буквы q (имитирует поведение одинарных кавычек) или qq (имитирует поведение двойных кавычек):

## Пример использования альтернативных кавычек

```
$pi      = 3.141592654;
$link1   = q{<a href="http://ru.wikipedia.org/wiki/Pi">$pi</a>};
print $link1;                                # <a
href="http://ru.wikipedia.org/wiki/Pi">$pi</a>
$link2   = qq{<a href="http://ru.wikipedia.org/wiki/Pi">$pi</a>};
print $link2;                                # <a
href="http://ru.wikipedia.org/wiki/Pi">3.141592654</a>
```

Ещё один тип кавычек в Perl — обратные кавычки. В них помещаются программы операционной системы, вывод которых может быть передан интерпретатору Perl. Выполнение программ всегда заканчивается символом новой строки — \n:

## Пример использования обратных кавычек для исполнения команд UNIX

```
$space    = `du -s -k /usr/local/bin`;
print $space;                                # 6264  /usr/local/bin

$sysuptime = `uptime`;
print $sysuptime;                            # 2:24  up 1 day,  9:05, 1 user,
load averages: 0.26 0.32 0.33
```

## [\[править\]](#)Массив

- [Массив](#) является упорядоченным списком [скаляров](#). Каждый элемент массива имеет порядковый индекс, с помощью которого к нему можно получить доступ. Нумерация элементов начинается с нуля, то есть первый элемент списка имеет индекс 0. Перед именем переменной-массива необходимо ставить знак '@', а для доступа к определенному элементу массива использовать знак '\$', так как определенный элемент массива является скаляром. Многомерные массивы можно смоделировать, помещая в список ссылки на другие списки.

```
@array = (1,2,3,4); # записали в массив @array четыре элемента
print $array[2]; # напечатать третий элемент
print @array[2,3]; # напечатать третий и четвертый элемент.
# Здесь используется @, так как результат операции среза массива тоже является массивом.
```

## [\[править\]](#)Хеш-таблица

- Хеш-таблица представляет собой [ассоциативный массив](#), позволяющий ассоциировать строку (называемую ключ) со скаляром (значение). Строка называется ключом, а скаляр в хеш-таблице значением. Перед именем переменной-списка необходимо ставить знак процента '%', а для доступа к определенному элементу массива необходимо ставить знак '\$'.

Хеш-таблицу можно инициализировать массивом или списком, состоящим из неограниченного числа последовательностей (ключ, значение).

```
%hash = (
    cat => 'kitten', # здесь => - это так называемая "длинная запятая", в
    этом примере
```

```

        dog => 'puppy',    # по функции полностью аналогичная обыкновенной ",",
за исключением
        cow => 'calf'      # того, что литералы слева неявно заключаются в
кавычки
    );
print $hash{'cat'}; #Напечатает kitten
print join("-", keys %hash) ; # Вывод всех ключей. Напечатает cat-dog-cow
print join("-", values %hash) ; # Вывод всех значений. Напечатает kitten-puppy-
calf
print join("-", %hash) ; # Напечатает cat-kitten-cow-calf-dog-puppy, так как в
списковом контексте хэш автоматически
# разворачивается в список (ключ, значение). Так как ключи в хэше не сортируются,
# пары каждый раз будут выводиться в произвольном порядке.

```

### [\[править\]](#)Функция

- Функция представляет собой фрагмент исполняемого кода. Функция всегда возвращает какое-либо значение. Если возвращаемое значение явно не указано оператором return, возвращается последнее вычисленное значение. Если в функции используется return без параметров, то в зависимости от контекста вызова функции возвращается неопределённое значение undef, пустой список или вообще ничего.

```

sub printTwo{
    print 2;
}

sub three{
    3;
}

$s = \&three;
print &$s; #Напечатает 3

$s = \&printTwo; # Запишем в скаляр $s ссылку на функцию - через эту ссылку она
будет вызвана в следующей строке.
print &$s; #Напечатает 21. `2` в коде функции, а `1` как значение, возвращаемое
функцией printTwo
print &printTwo #То же самое

```

### [\[править\]](#)Константа

- Константа представляет собой неизменяемое значение. Константа не является встроенным типом языка и эмулируется при помощи функций. Предпочтительно задавать константы с помощью стандартного модуля constant, чтобы в будущем не зависеть от возможных изменений в реализации констант. Использование:

```

use constant MY => 2;
print MY;

```

### [\[править\]](#)Запись таблицы символов

- Запись таблицы символов — специальный тип, в котором хранятся ссылки на все переменные других типов с таким названием, а также на символы, которые часто использует для хранения файловый дескриптор, представляющих собой указатель на файл, устройство или PIPE канал, открытые для записи, чтения или для записи и чтения.

Использование:

```
$s = <STDIN>; #Читаем одну строку из дескриптора STDIN (стандартный поток ввода);  
скалярный контекст.  
@values = <FILE>; #Читаем все строки из некоторого открытого и связанного с  
дескриптором FILE файла; списковый контекст.  
print STDOUT $s; #Печатаем в STDOUT (стандартный поток вывода)
```

Более информативный пример, в котором `typeglob` используется для передачи файлового дескриптора в функцию:

```
my $file = 'file.txt'; #имя файла, который читаем  
open FH, $file; #открываем файл  
my $text = _parseFile(*FH); #передаем файловый дескриптор в функцию  
print $text;  
close FH; #обязательно закрываем файловый дескриптор  
sub _parseFile  
{  
    my $fh = shift; #получаем файловый дескриптор  
    my $text = join(' ', <$fh>); #построчно читаем из файлового дескриптора, в  
строку и помещаем в переменную $text  
    return $text;  
}
```

В последних версиях Perl появилась возможность хранить файловые дескрипторы в скалярах, и в новых программах предпочтительно пользоваться именно этим способом.

Также таблицы символов используются для связи двух имён переменной с одним значением в памяти, создавая синонимы, которые могут быть использованы для доступа и модификации значения точно также как и оригинальное имя. Эта возможность является основой системы загружаемых модулей, [классов и объектов](#) в Perl.

## [\[править\]](#) **Классы и объекты**

Объект в Perl представляет собой просто ссылку, но не любую, а связанную с определенным классом(пакетом). Для связи используется функция `bless`. Методы в Perl являются обычными подпрограммами. Конструктор — это просто подпрограмма, возвращающая ссылку.

Так же существуют множество расширения для системы ООП, позволяющие менять до неузнаваемости синтаксис и использовать новый «синтаксический сахар», в числе которых стоит назвать `cpan`-модули `Moose` и/или `Mouse`, `MooseX::Declare`. `Moose` используется многими проектами и крупными компаниями, такими как [BBC](#) или [Cisco](#)<sup>[20]</sup>.

## [\[править\]](#) **Регулярные выражения**

Важной частью Perl являются [регулярные выражения](#). Благодаря этому Perl хорошо подходит для обработки текстов. Бóльшая часть работы с регулярными выражениями производится при помощи операторов `=~`, `m//` и `s///`.

Оператор `m//` используется для проверки на совпадение. В простейшем случае результат выражения `$x =~ m/abc/` будет истинным, если и только если строка `$x` будет соответствовать регулярному выражению `abc`. Например:

| Пример                                    | Значение                                                                                                                                                                                                    |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\$x =~ /abc/</code>                 | Строка <code>\$x</code> содержит (под)строку «abc». Начальная буква «m» оператора при использовании <code>//</code> может быть опущена.                                                                     |
| <code>\$x =~<br/>m/a(.{1,3})c/</code>     | Строка <code>\$x</code> содержит букву «a», затем от одного до трех любых символов кроме символа перевода строки «\n», и затем букву «c». Найденные символы будут сохранены в переменную <code>\$1</code> . |
| <code>\$x =~<br/>m{^p(erl ugs)\$}i</code> | Строка <code>\$x</code> строго равна «perl» или «pugs» без учета регистра. Так же, вместо <code>//</code> регулярное выражение окружено в <code>{ }</code> .                                                |

Поиск и замена выполняются при помощи оператора `s///`. Конструкция `$x =~ s/abc/def/;` заменит первое вхождение регулярного выражения `abc` на строку `def`.

| Пример                                                        | Значение                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\$x =~ s/abc/def/g;</code>                              | Все вхождения (на что указывает флаг <code>/g</code> — global) подстроки «abc» в <code>\$x</code> будут заменены на «def».                                                                                                                                                             |
| <code>\$x =~<br/>s/a(.{1,3})c/!\$1!/;</code>                  | Первое вхождение в <code>\$x</code> буквы «a», затем от одного до трех любых символов кроме символа перевода строки «\n», и затем буквы «c» будет заменено на эти символы между «a» и «c», окружённые «!». Например, «syntactic» станет «synt!cti!».                                   |
| <code>\$x =~<br/>s{^p(erl ugs)}{"P" .<br/>lc \$1}iegi;</code> | Здесь показан пример использования модификатора <code>/e</code> , указывающего на то, что вместо строки замены будет написан код, результат выполнения которого надо использовать. Все вхождения «perl» или «pugs» в любом регистре будут заменены на «Perl» и «Pugs», соответственно. |

С появлением динамических регулярных выражений (`??{ код Perl }`) и конструкции для включения в регулярное выражение исполняемого кода Perl (`{ код Perl }`) возможности по поиску и замене стали практически безграничными. К примеру, стал возможен поиск конструкций произвольного уровня вложенности.

Регулярные выражения Perl настолько популярны, что они включены напрямую в другие языки, такие как [PHP](#) и [JavaScript](#), а также существуют подключаемые библиотеки, реализующие использование выражений в компилируемых языках.

[\[править\]](#) Программы из одной строки

Интересной и часто используемой возможностью Perl являются так называемые one-liners — программы из одной строки, обычно задаваемые прямо в строке вызова интерпретатора с помощью ключа `-e`.

Эта программа напечатает [простые числа](#):

```
perl -wle '(1 x $_) !~ /^(1|((11+)\3+))$/ && print while ++ $_'
```

Этот пример напечатает только уникальные строки файла file, ключ `-n` автоматически заворачивает строку обработки в цикл, который проходит по всем строкам в файлах, указанных после текста программы:

```
perl -ne '$_ = $_; print' file
```

Примечание: для запуска предыдущей команды под [Windows](#) замените в ней апострофы `'` на двойные кавычки `"`.

В этих примерах Perl напоминает своей краткостью и непонятностью с первого взгляда язык [J](#).

Также одним из подобных примеров является вызвавшая большой

резонанс [\[источник не указан 175 дней\]](#) программа [\[21\]](#), на самом деле являющаяся [патчем](#)

[Бармина](#) (замаскированной командой рекурсивного удаления всех файлов):

```
echo "test... test... test..." | perl -e '$??s;;s;s;;$?:s;;]=>%-{-|}<&|`{;;y; -/:-@[-`{-};`-{" -;;s;;$_;see'
```

`echo "test... test... test..."` выполнение этой команды не влияет на работу и добавлено, скорее всего, для усыпления бдительности. То, что происходит в остальном коде — совсем не очевидно из-за преднамеренно запутанного написания. В данной строчке записано всего три последовательно выполняемых команды. Запишем команду следующим образом:

```
$? ? s;;s;s;;$?: : s;;]=>%-{-|}<&|`{ ; ;
y; -/:-@[-`{-};`-{" -;; ;
s;;$_;see
```

Первая конструкция анализирует переменную `$?` — код возврата предыдущей команды. Так как перед выполнением этой конструкции дочерних процессов не создавалось, `$?` будет содержать 0, и выполнена будет вторая «ветка» — `s;;]=>%-{-|}<&|`{`. Эта команда, в свою очередь, заменяет строку в переменной-аккумуляторе `$_` на `=]=>%-{-|}<&|`{` (первый символ после `s` устанавливает ограничитель параметров этого оператора, и хотя традиционно используются слэш `'` или `|`, для неясности в этой конструкции используется ограничитель `';`).

Вторая команда транслирует содержимое «аккумулятора» по достаточно сложным правилам. В левой части указано четыре диапазона символов, в правой — один. Если раскрыть эти диапазоны, получим следующее соответствие:

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}
`abcdefghijklmnopqrstuvwxyz{/" -
```

В результате содержимое `$_` принимает вид

```
system"rm -rf /"
```

Третья же команда дважды (как инструктирует флаг `ee`) «вычисляет» содержимое аккумулятора — [вышеуказанную деструктивную команду](#) — и пытается заменить пустую строку в аккумуляторе на результат вычисления.

## [\[править\]](#) Языки программирования и оболочки со схожими технологиями

---

Perl — далеко не единственный язык, дающий богатые возможности в обработке текста. Языки программирования [РЕФАЛ](#), [Icon](#) и [Снобол](#) предоставляют возможность более полно использовать метод программирования «сопоставления с образцом», частным случаем которого и являются регулярные выражения.

Существует также полноценный [шелл](#), использующий в качестве основы язык Perl. Называется он [psh](#) и даёт возможность смешивать команды обычного шелла и самого Perl'a<sup>[22]</sup>.

## [\[править\]](#) Поэзия в Perl

---

На Perl можно даже писать стихотворения. Одно из таких стихотворений под названием «Black Perl» («Чёрная жемчужина» или «Чёрный перл») было упомянуто Ларри Уоллом в первоапрельском письме в [Usenet](#). Оно было написано изначально для Perl 3, и по словам самого Ларри, он испытал моральное удовлетворение, когда стихотворение не прошло синтаксический анализатор в Perl 5.

```
BEFOREHEAD: close door, each window & exit; wait until time.
open spellbook, study, read (scan, select, tell us);
write it, print the hex whole each watches,
reverse its length, write again;
kill spiders, pop them, chop, split, kill them.
unlink arms, shift, wait & listen (listening, wait),
sort the flock (then, warn the "goats" & kill the "sheep");
kill them, dump qualms, shift moralities,
values aside, each one;
die sheep! die to reverse the system
you accept (reject, respect);
next step,
kill next sacrifice, each sacrifice,
wait, redo ritual until "all the spirits are pleased";
do it ("as they say").
do it(*everyone**must**participate**in**forbidden**s*e*x*).
return last victim; package body;
exit crypt (time, times & "half a time") & close it,
select (quickly) & warn your next victim;
AFTERWORDS: tell nobody,
wait, wait until time;
wait until next year, next decade;
sleep, sleep, die yourself,
die at last
```

## [\[править\]](#) Редакторы

---

- [Padre IDE \(англ.\)](#)
- [PSPad](#)
- [ActiveState Komodo](#)
- [jEdit](#)
- [EPIC — Perl Editor and IDE for Eclipse](#)
- [Geany](#)
- [IDM Computer Solutions, Inc. — UltraEdit & UESudio](#)
- [Notepad++](#)
- [Gedit](#)
- [SciTE](#)
- [Emacs](#)
- [Vim](#)

**ASP.NET** — технология создания [веб-приложений](#) и [веб-сервисов](#) от компании [Майкрософт](#). Она является составной частью платформы [Microsoft .NET](#) и развитием более старой технологии [Microsoft ASP](#). На данный момент последней версией этой технологии является **ASP.NET 4.5**<sup>[1]</sup>.

ASP.NET внешне во многом сохраняет схожесть с более старой технологией ASP, что позволяет разработчикам относительно легко перейти на ASP.NET. В то же время внутреннее устройство ASP.NET существенно отличается от ASP, поскольку она основана на платформе [.NET](#) и, следовательно, использует все новые возможности, предоставляемые этой платформой.

ASP.NET не является платформой.

## История ASP.NET

---

После выпуска сервера [Internet Information Services](#) 4.0 в 1997 году, компания [Microsoft](#) начала исследовать возможность новой модели веб-приложения, которая удовлетворит жалобы на ASP, особенно связанные с отделением оформления от содержания, и которая позволит писать «чистый» код<sup>[2]</sup>. Работа по разработке такой модели была поручена Марку Андерсу, менеджеру команды IIS, и [Скотту Гатри](#), поступившему на работу в [Microsoft](#) в 1997. Андерс и Гатри разработали первоначальный проект в течение двух месяцев, и Гатри написал код первоначального прототипа во время рождественских каникул 1997 года.<sup>[3]</sup>

Первоначальный проект назывался «XSP»; Гатри объяснил в интервью 2007 года что, «всегда спрашивают, что означает буква X. В то время она ничего не значила. XML начинается с нее; XSLT начинается с нее. Все клевое начинается с X, поэтому мы его так и назвали.»<sup>[2]</sup> Прототип XSP был написан на [Java](#), так как на тот момент у [Microsoft](#) не было Java-подобной технологии. В то время уже предполагалось (как дальше выяснилось основательно), что лицензирование [Java](#) для [Microsoft](#) не будет продлено в 2003 году (в 2003 истек срок выданной [Sun Microsystems](#) лицензии). В 1999 было решено построить новую платформу на основе [Common Language Runtime](#)(CLR), так как в нем как и в [Java](#) наличествовало программирование по принципам [ООП](#), [Сборка мусора](#) и другие возможности<sup>[4]</sup>. Гатри описал это решение как «огромный риск», так как успех новой разработки был связан с успехом CLR, которая, как и XSP, находилась на ранней стадии разработки.

## [\[править\]](#) Принципы ASP.NET

---

Хотя ASP.NET берёт своё название от старой технологии Microsoft ASP, она значительно от неё отличается. Microsoft полностью перестроила ASP.NET, основываясь на [Common Language Runtime](#) (CLR), которая является основой всех приложений Microsoft .NET. Разработчики могут писать код для ASP.NET, используя практически любые языки программирования, входящие в комплект [.NET Framework](#) ([C#](#), [Visual Basic.NET](#) и [JScript .NET](#)). ASP.NET имеет преимущество в скорости по сравнению со скриптовыми технологиями, так как при первом обращении код компилируется и помещается в специальный [кэш](#), и впоследствии только исполняется, не требуя затрат времени на [парсинг](#), [оптимизацию](#), и т. д.

## [\[править\]](#) Преимущества ASP.NET перед ASP

---

- Компилируемый код выполняется быстрее, большинство ошибок отлавливается ещё на стадии разработки
- Значительно улучшенная обработка ошибок времени выполнения, с использованием блоков try..catch
- Пользовательские элементы управления (controls) позволяют выделять часто используемые шаблоны, такие как меню сайта
- Использование метафор, уже применяющихся в Windows-приложениях, например, таких как элементы управления и события
- Расширяемый набор элементов управления и библиотек классов позволяет быстрее разрабатывать приложения
- ASP.NET опирается на многоязыковые возможности .NET, что позволяет писать код страниц на VB.NET, Delphi.NET, Visual C#, J# и т. д.
- Возможность кэширования всей страницы или её части для увеличения производительности
- Возможность кэширования данных, используемых на странице
- Возможность разделения визуальной части и бизнес-логики по разным файлам («code behind»)
- Расширяемая модель обработки запросов
- Расширенная событийная модель
- Расширяемая модель серверных элементов управления
- Наличие master-страниц для задания шаблонов оформления страниц
- Поддержка [CRUD](#)-операций при работе с таблицами через GridView
- Встроенная поддержка [AJAX](#)
- ASP.NET имеет преимущество в скорости по сравнению с другими технологиями, основанными на скриптах.

Здесь можно привести определённые сравнения. Так, ASP — производная от Win32, XML и HTML; PHP — от XML, HTML, Java и CDI, тогда ASP.NET — от HTML и .NET (XML и XAML соответственно). При этом, если обычно Rich Media Application создают при помощи Flash, теперь это делается с помощью модуля Silverlight, так же через сам ASP.NET. ASP.NET — богатейшая <sup>[\[источник не указан 599 дней\]](#)</sup> среда для разработки и развёртывания веб-ресурсов. В ASP.NET можно работать с любым .NET языком, вплоть до Managed C++ и Visual Basic, что позволяет не задумываться о переходе на C#.

## [\[править\]](#) Расширения

---

[Корпорация Майкрософт](#) выпустила несколько расширений для ASP.NET:

- [ASP.NET AJAX](#)
- [ASP.NET MVC Framework](#)

## **В2. Обобщённая структура сети Интернет**

В архитектуре [Internet](#) отдельные сети (ЛВС, региональные и глобальные) соединяются друг с другом специальными устройствами – маршрутизаторами IP-пакетов.

**Определение.** Устройства объединения сетей в рамках Internet называются IP-шлюзами или IP-маршрутизаторами, или Router.



LAN – локальная вычислительная сеть;

MAN – региональная ИВС;

WAN – глобальная ИВС;

WS (Work Station) – рабочая станция ЛВС;

FS (File Server) – файл-сервер;

Host – узловая машина (компьютер, который подключен к сети в качестве узла);

Router – IP-маршрутизатор.

Шлюз подключается к двум или более сетям, каждая из которых воспринимает этот шлюз как хост-ЭВМ. Поэтому шлюз имеет физический интерфейс и специальный IP-адрес в каждой из подключаемых сетей.

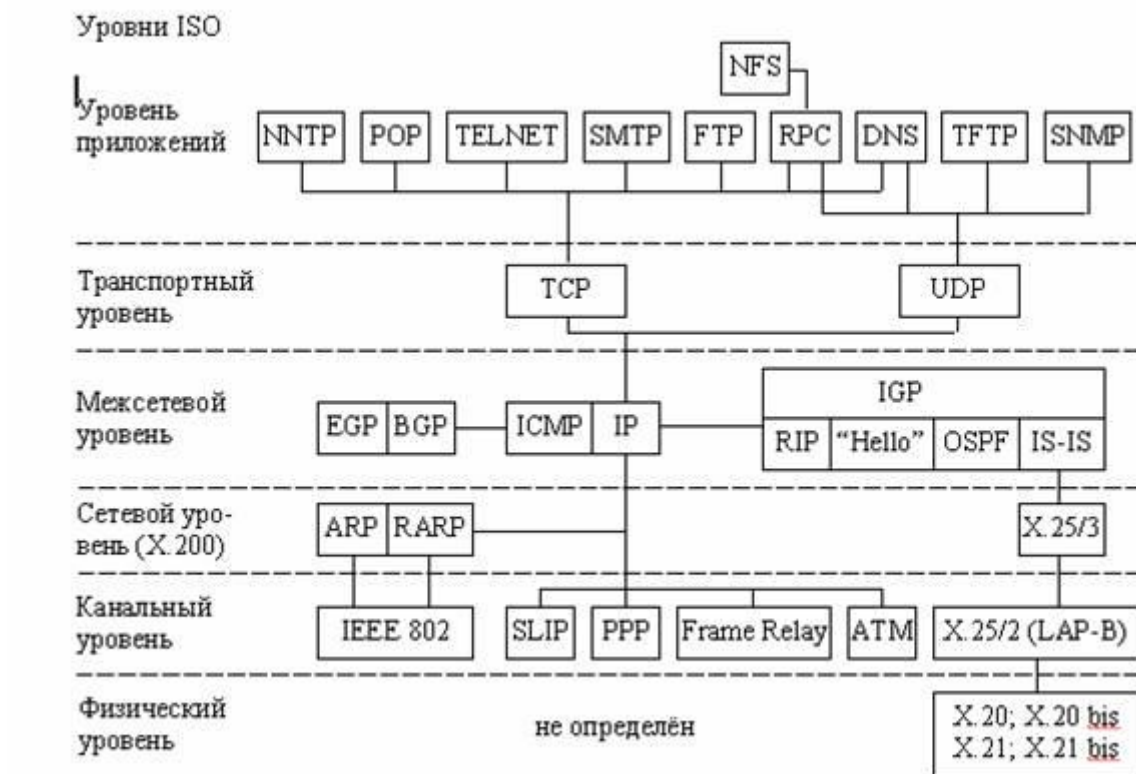
Передача пакетов требует от шлюза определение IP-адреса следующего шлюза или, на последнем участке, IP-адреса хост-машины, которой направляется IP-пакет.

Функция шлюза, которая обычно называется маршрутизацией, основана на анализе специальных маршрутных таблиц (матриц маршрутов), которые находятся в специальной базе данных. База данных в каждом из шлюзов должна постоянно обновляться, чтобы отражать текущую топологию сети Internet.

**Маршрут** – это последовательность маршрутизаторов, которые проходит пакет от отправителя до пункта назначения.

В основе функционирования сети Интернет заложены протоколы TCP/IP.

Основные протоколы семейства TCP/IP приведены на рис. 2. Согласно рис. 2 одни протоколы верхнего уровня (например, Telnet и FTP) зависят от TCP, а другие (например, TFTP и RPS) — от UDP. Большинство из них используют только один из этих транспортных протоколов, но некоторые (например, DNS) — оба.



="" p="">

Рис. 2. Основные протоколы семейства TCP/IP:

NFS — Network File System — сетевая файловая система;

NNTP — Network News Transfer Protocol — протокол сетевой передачи новостей;

POP — Post Office Protocol — протокол почтового отделения;

TELNET — Terminal Networking — протокол и программные средства, позволяющие подключаться к удалённой машине и работать с ней через эмулируемый терминал;

SMTP — Simple Mail Transfer Protocol — простой протокол электронной почты;

FTP — File Transfer Protocol — протокол передачи файлов;

RPC — Remote Procedure Call — вызов удалённых процедур;

DNS — Domain Name Service — служба именования доменов;

TFTP — Trivial File Transfer Protocol — простейший протокол передачи файлов;

SNMP — Simple Network Management Protocol — простой протокол управления сетью;

TCP — Transmission Control Protocol — протокол управления передачей данных;

UDP — User Datagram Protocol — протокол пользовательских дейтаграмм;

EGP — Exterior Gateway Protocol — протокол внешней маршрутизации;

BGP — Border Gateway Protocol — протокол граничных маршрутизаторов;

IP — Internet Protocol — межсетевой протокол;

ICMP — Internet Control Message Protocol — межсетевой протокол управляющих сообщений;

IGP — Interior Gateway Protocol — внутренний протокол маршрутизации;

RIP — Routing Information Protocol — протокол для передачи маршрутной информации;

“Hello” — реализация протокола внутренней маршрутизации;

OSPF — Open Shortest Path First — открытый протокол предпочтения кратчайшего пути;

IS-IS — Intermediate System to Intermediate System Protocol — протокол маршрутизации, выполняющий маршрутизацию данных IP и МОС;

ARP — Address Resolution Protocol — протокол преобразования адресов;

RARP — Reverse Address Resolution Protocol — протокол обратного преобразования адресов;

X.25/3 — протокол пакетного уровня сети передачи данных;

IEEE 802 — стандарт локальных сетей;

SLIP — Serial Line Internet Protocol — межсетевой протокол для последовательного канала;

PPP — Point-to-Point Protocol — протокол “точка-точка”;

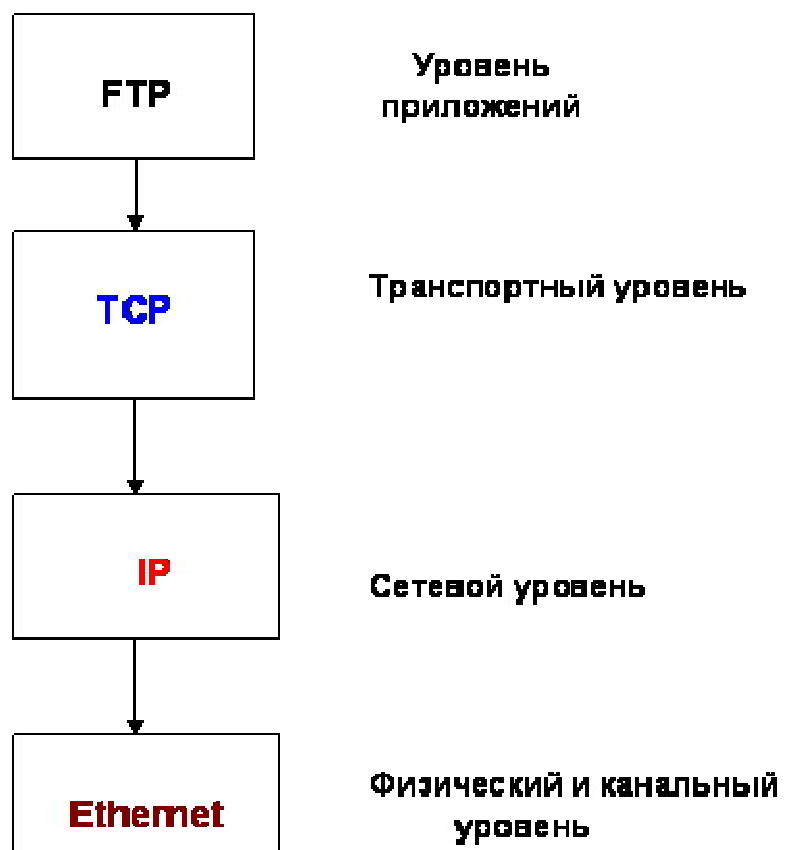
Frame Relay — сетевой механизм для быстрой пересылки кадров;

ATM — Asynchronous Transfer Mode — режим асинхронной пересылки;

X.25/2 (LAP-B) — протокол для управления передачей кадров (Link Access Procedures Balanced — сбалансированные процедуры доступа к каналу);

X.20; X.20 bis — сопряжение оборудования обработки данных с асинхронными модемами;

X.21; X.21 bis — сопряжение оборудования обработки данных с синхронными модемами.



="" p="">

<

Рис. 3. Пример цепочки протоколов TCP/IP

Данные передаются в пакетах. Пакеты имеют заголовок, который содержит служебную информацию. Данные более верхних уровней вставляются в пакеты нижних уровней.

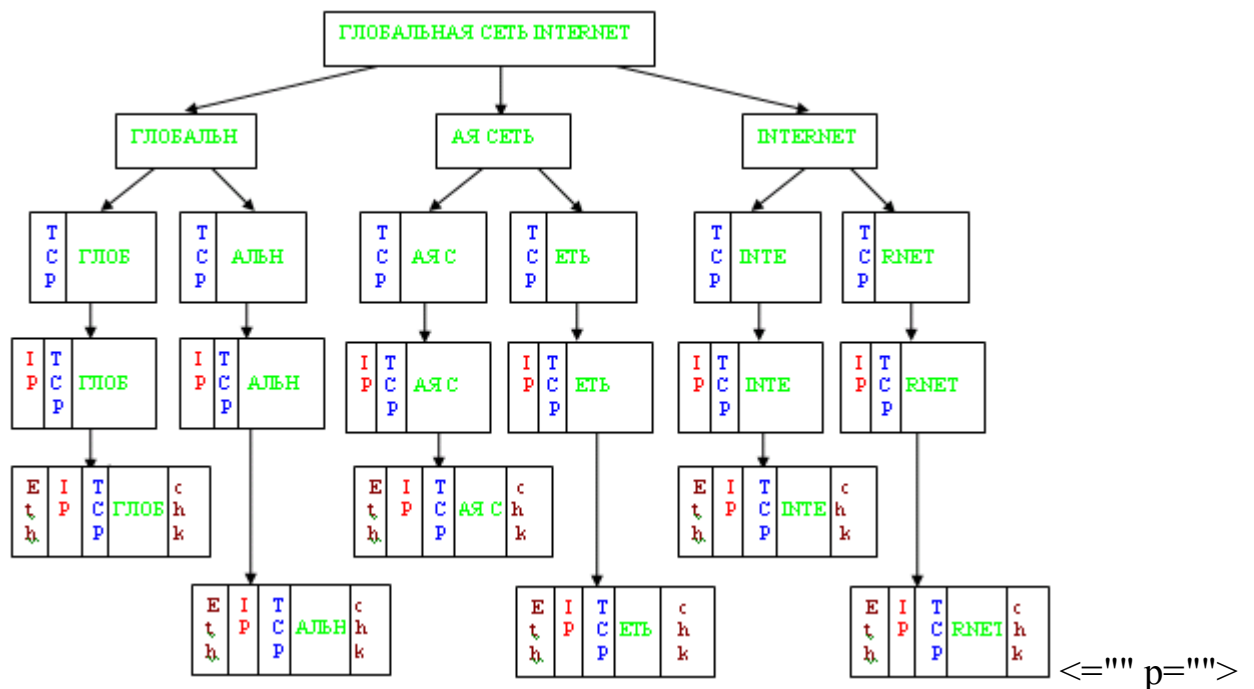


Рис. 4. [Передача сообщений в сети Internet на основе механизма инкапсуляции \(encapsulation\)](#)

### В3. Стек протоколов ТСП/ІР

ТСР/ІР – собирательное название для набора (стека) сетевых протоколов разных уровней, используемых в Интернет. Особенности ТСР/ІР:

- открытые стандарты протоколов, разрабатываемые независимо от программного и аппаратного обеспечения;
- независимость от физической среды передачи;
- система уникальной адресации;

- стандартизированные протоколы высокого уровня для распространенных пользовательских сервисов.

Стек протоколов TCP/IP делится на 4 уровня:

- прикладной,
- транспортный,
- межсетевой,
- физический и канальный.

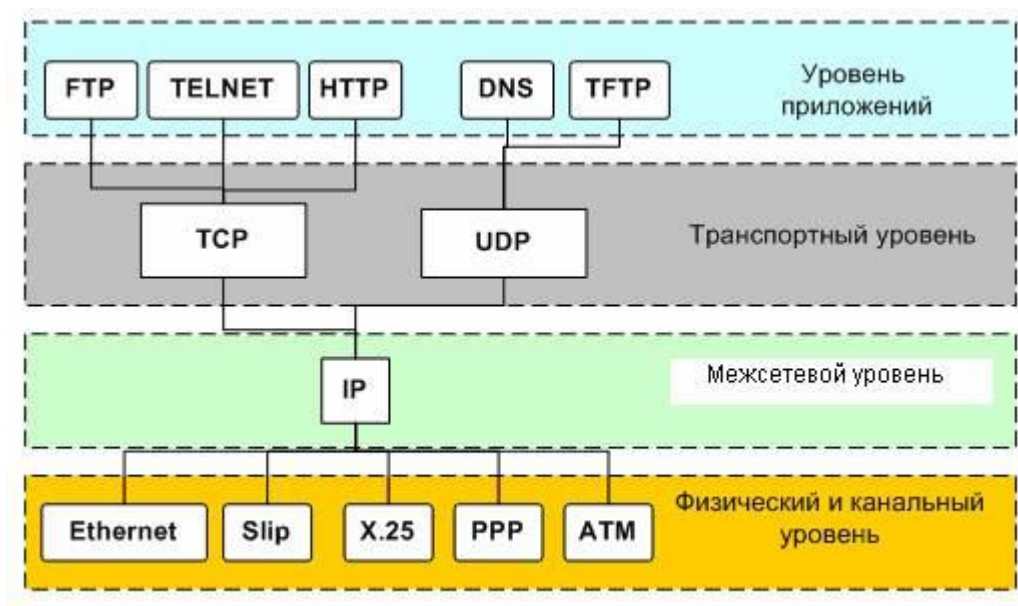


Рис. 5. Пример стека протоколов TCP/IP

Каким образом мы попадаем со своего компьютера на удаленный сервер?

Все наши компьютеры объединены в локальную сеть, и имеют локальную IP-адресацию. Пакеты с такой адресацией "путешествовать" в глобальной сети не смогут, т.к. маршрутизаторы их не пропустят.

Поэтому существует шлюз, который преобразовывает пакеты с локальными IP-адресами, давая им свой внешний адрес. И дальше пакеты путешествуют с адресом шлюза.

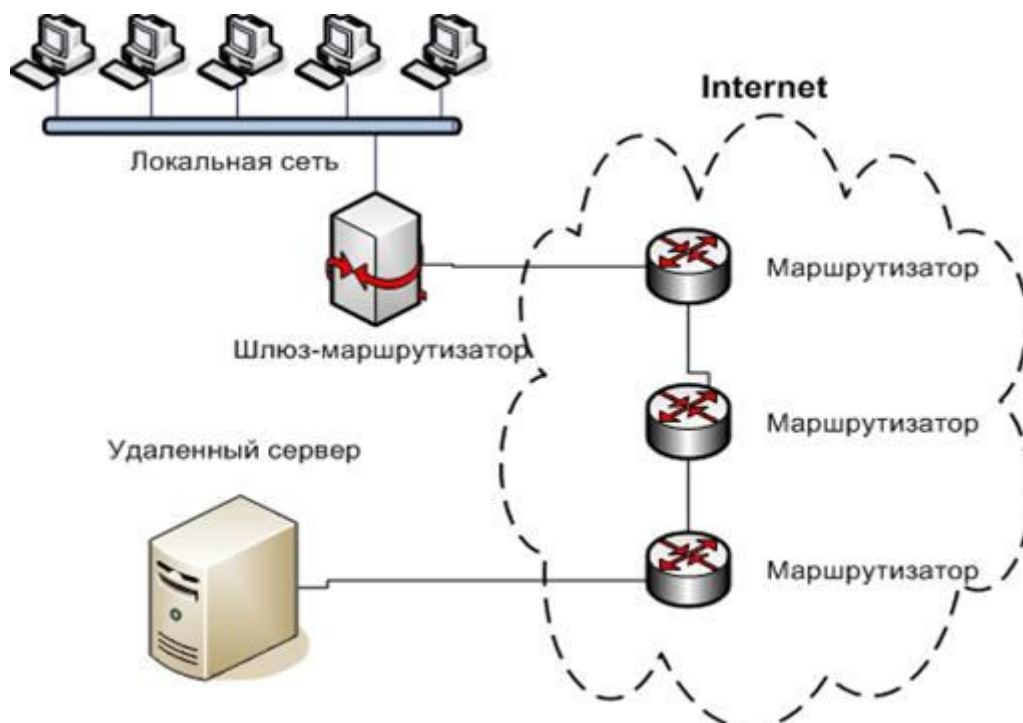


Рис. 6. [Схема прохождения пакетов из локальной сети к серверу](#)

Локальных сетей слишком много, поэтому реально объединяют автономные системы.

**Автономная система** (AS - autonomous system) – сеть, находящаяся под одним административным контролем, это может быть несколько компьютеров или большая сеть.

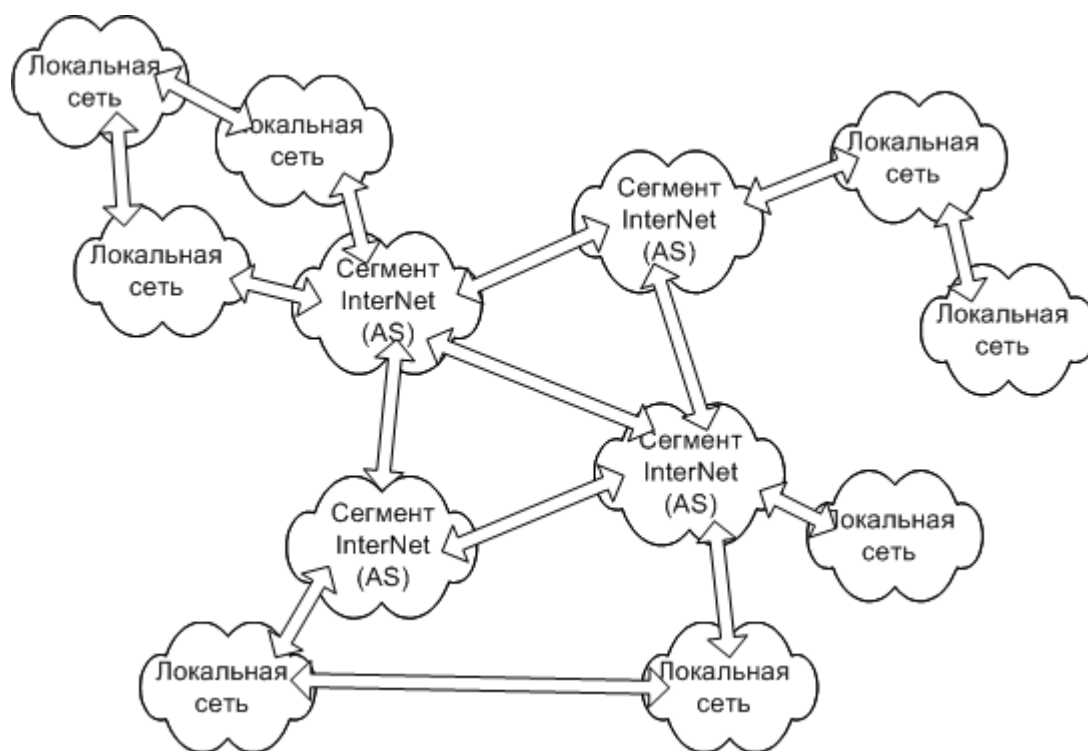
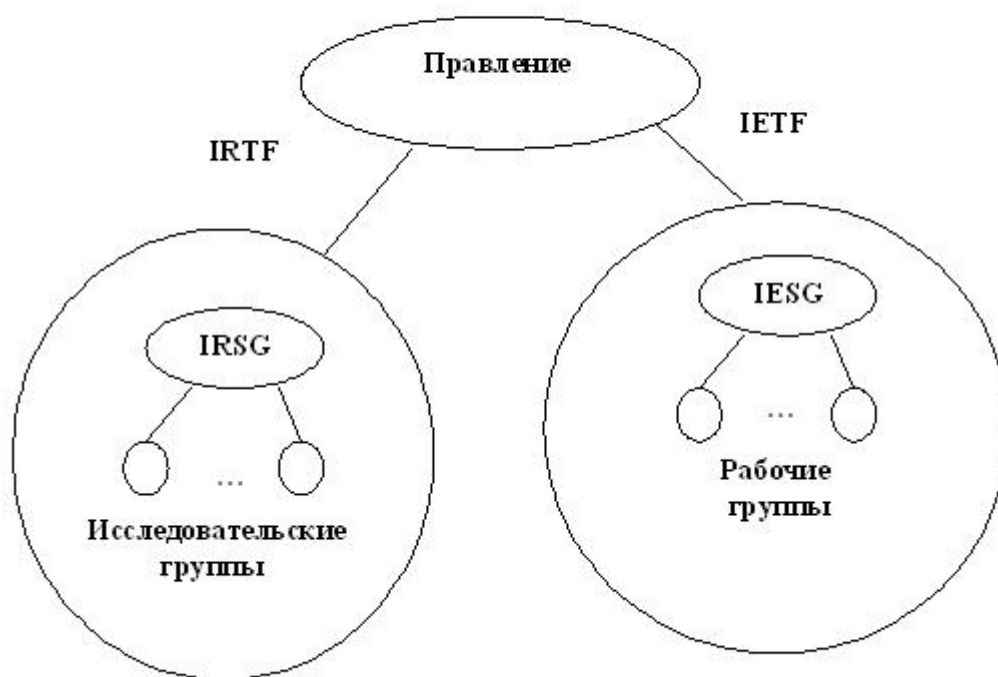


Рис. 7. Схема объединения отдельных сетей в общую составную сеть

## В4. Архитектурный совет Internet



## Рис. 8. Структура архитектурного совета Internet

IRTF (Internet Research Task Force) – служба по вопросам исследования сети Интернет.

IETF (Internet Engineering Task Force) – служба инженерной поддержки сети Интернет.

IRSG (Internet Research Steering Group) – руководящая группа по вопросам исследования сети Интернет.

IESG (Internet Engineering Steering Group) – руководящая группа по разработке сети Интернет.

**RFC (Request for Comments)** – официальная документация по Internet. RFC-документы можно найти по адресу <http://www.rfc-editor.org/> или <http://www.ietf.org/rfc.html> .

Все разработчики должны придерживаться этой документации.

В качестве одного из первых составителей и редакторов большой серии документов RFC был [Джонатан Постел](#).

[Это интересно...](#)

### **Контрольные вопросы:**

1. Дайте определение понятия сети Internet.
2. На базе каких сетей возникла Internet?
3. Что понимают под структурой сети Internet?
4. Опишите обобщённую структуру сети Internet.
5. Объясните структуру стека протоколов TCP/IP.

### **Материал для самостоятельной подготовки**

## 1. Сравнительная оценка сетевых архитектур ISO и TCP/IP

Главным направлением развития современных информационно-вычислительных сетей (ИВС) являются их глобализация и объединение (интеграция). Это приводит к расширению ИВС, совместному использованию программного обеспечения (ПО), объединению различных сетей и т.п.

В реальных сетях используется множество сетевых архитектур, таких как TCP/IP, IPX/SPX, XNS XEROX, Apple Talk, SNA, Banyan VINES, ISO, 3COM, DECnet и ряд других.

Однако наибольшее распространение получили два подхода – архитектура TCP/IP американского научно-исследовательского центра DARPA и архитектура сети на базе стандарта ISO. Принципиальные отличия этих архитектур вытекают из учёта качества используемых каналов связи. Так, архитектура TCP/IP ориентирована на применение достаточно хороших каналов связи с низким коэффициентом ошибок (порядка  $10^{-5}$ ), в то время как архитектура ISO допускает использование каналов с вероятностью ошибки порядка  $10^{-3}$ .

Так как основная задача ИВС общего пользования состоит в организации взаимодействия разнородных пользователей на значительных территориях, то главными требованиями к сетевой архитектуре являются:

наличие мощной, открытой и гибкой системы адресации, позволяющей обеспечить обслуживание значительного количества пользователей;

высокая эффективность передачи полезной информации в сети как по времени, так и по верности доставки;

высокая степень адаптации к изменяющимся внешним условиям (неисправности, подключение новых ресурсов или пользователей).

Стек основных протоколов сетевых архитектур ISO и TCP/IP представлен в табл. 1.1.

Можно выделить следующие существенные отличия данных архитектур:

Архитектура ISO предусматривает жёсткий набор протоколов на всех уровнях модели, когда на каждом уровне между взаимодействующими объектами сначала устанавливается логическая связь, а уже затем передаются данные. При этом сверху до низу сохраняется последовательность передачи протокольных единиц (блоков, фрагментов,

пакетов, кадров) и предпринимаются специальные меры для сохранения целостности этих порций данных. В случае потери или искажения протокольной единицы на каждом уровне (кроме физического) осуществляются перезапрос и повторная передача искажённой протокольной единицы.

Архитектура TCP/IP предусматривает возможность ветвления протоколов и даже добавление новых. За целостностью данных следит транспортный уровень (протокол TCP) либо сам пользователь (протокол UDP).

Таблица 1

| Уровни стандарта ISO | Стек протоколов стандарта ISO | Стек протоколов TCP/IP                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7. Прикладной        | Набор протоколов              | <pre> graph TD     subgraph Top [Набор протоколов]         direction LR         A(Набор протоколов)         B(Набор протоколов)     end     A &lt;--&gt; TCP     B &lt;--&gt; UDP     TCP &lt;--&gt; IP     UDP &lt;--&gt; IP     style Top fill:none,stroke:none     style TCP fill:none,stroke:none     style UDP fill:none,stroke:none     style IP fill:none,stroke:none         </pre> |
| 6. Представительный  | X.226                         |                                                                                                                                                                                                                                                                                                                                                                                             |
| 5. Сеансовый         | X.225                         |                                                                                                                                                                                                                                                                                                                                                                                             |
| 4. Транспортный      | X.224                         |                                                                                                                                                                                                                                                                                                                                                                                             |
| 3. Сетевой           | X.25/3                        |                                                                                                                                                                                                                                                                                                                                                                                             |
| 2. Канальный         | LAP-B(X.25/2)                 | Произвольный                                                                                                                                                                                                                                                                                                                                                                                |
| 1. Физический        | X.21                          | Произвольный                                                                                                                                                                                                                                                                                                                                                                                |

Различия в идеологии построения сетевых архитектур порождают существенные различия механизма передачи данных на всех уровнях стандарта ISO за исключением физического и канального, где могут применяться протоколы LAP-B и X.21, но могут и другие. Основные отличия в алгоритме передачи данных состоят, во-первых, в **идеологии защиты от ошибок**, и, во-вторых, в реализации **режима коммутации пакетов (КП)**.

Рассмотрим сначала **методы борьбы с ошибками**.

Вопросам защиты данных от ошибок и сбоев уделено много внимания. Для этого выделяется второй (канальный) уровень. Обнаружение ошибок выполняется с помощью мощного помехоустойчивого кода типа БЧХ (Рек. V.42) с минимальным кодовым расстоянием  $d_0=5$ , что позволяет обнаруживать любую 4-х кратную ошибку. Исправление ошибок

выполняется с помощью алгоритмов с обратной связью – РОС-ОЖ или (чаще) РОС-НП. Для борьбы со вставками и выпадениями кадров используются тайм-аут и циклическая нумерация кадров. На сетевом уровне обеспечиваются нумерация пакетов и их перезапрос. Всё это позволяет использовать передающую среду практически любого качества, однако платой за это является высокая степень вносимой избыточности, т.е. падение реальной скорости передачи информации.

В архитектуре TCP/IP первый и второй уровни вообще не оговорены, т.е. передача может вестись даже без защиты от ошибок. Повышение верности возложено на транспортный протокол TCP. Если используются хорошие каналы, например, волоконно-оптические линии связи (ВОЛС), то на транспортном уровне используется протокол UDP, где не предусмотрена защита от ошибок. В этом случае обнаружение и исправление ошибок осуществляются на прикладном уровне специальными программами пользователя. Такой подход становится понятным, т.к. архитектура TCP/IP первоначально была реализована в сети ARPANET, где использовались выделенные высокоскоростные каналы.

Рассмотрим различия в способах **коммутации пакетов**, т.е. в реализации 3-го уровня ISO.

В архитектуре ISO за маршрутизацию (доставку пакетов по адресу) отвечает третий (сетевой) уровень (Рек. Х.25). Предусматривается создание виртуальных соединений или каналов от источника до получателя, а затем по этому соединению передаются пакеты. Такой режим называется виртуальным режимом КП и по принципам напоминает традиционную коммутацию каналов (КК). В архитектуре TCP/IP реализуется другой подход, называемый дейтаграммным режимом КП. тот режим резко упрощает задачу маршрутизации, но порождает проблему сборки сообщений из пакетов, т.к. пакеты одного сообщения могут доставляться по разным маршрутам и поступать к получателю в разное время. Дейтаграммный режим КП по принципам напоминает коммутацию сообщений (КС).

Проведём сравнения виртуального и дейтаграммного методов КП по следующим характеристикам:

- установление соединения;
- адресация;
- процедура передачи пакета по сети;

- управление входным потоком сообщений;
- эффективность использования сетевых ресурсов.

**Установление соединения.** При виртуальной КП до передачи сообщения устанавливается логическое соединение между взаимодействующими объектами транспортного уровня (а возможно и более высоких уровней ISO). Этот логический канал запоминается в маршрутных таблицах всех центров коммутации пакетов (ЦКП), которые участвуют в соединении. Пакеты передаются только по установленному логическому каналу, поэтому порядок их следования при этом не нарушается.

При дейтаграммой КП логического соединения не устанавливается, поэтому пакеты одного сообщения передаются по тем маршрутам, которые оптимальны в данный момент, т.е. возможно разными маршрутами. Проблема сборки сообщения из пакетов решается на транспортном уровне.

**Адресация.** При виртуальном режиме КП полный адрес объекта-получателя передаётся только при установлении логического соединения, т.е. с первым пакетом. Получив этот пакет, объект-получатель извещает отправителя о согласии на проведение сеанса связи (или несогласии). Создаётся логическое соединение, и передаются остальные пакеты, содержащие только номер логического канала.

При дейтаграммном режиме КП каждый передаваемый пакет обязательно должен содержать полный адрес получателя (и отправителя) и номер пакета в сообщении.

**Процедура передачи пакета по сети.** Виртуальный режим КП предусматривает выделение специальной базовой сети передачи данных (ПД) и передачу пакетов в этой сети ПД по готовому логическому каналу, создаваемому по инициативе транспортного уровня.

При дейтаграммном режиме каждый пакет передаётся по разным маршрутам, что позволяет эффективнее использовать сетевые ресурсы, т.к. в больших сетях загрузка каналов меняется очень быстро, поэтому маршрут доставки желательно корректировать чаще. В данном случае можно построить глобальную сеть без выделения отдельной базовой сети ПД.

**Управление входным потоком сообщений.** При виртуальном режиме КП управление потоком входящих сообщений (но не пакетов) возможно

лишь на входе виртуального канала, т.е. на конкретном центре коммутации пакетов для данного сообщения.

Дейтаграммный режим КП является более гибким и позволяет управлять входящим потоком сообщений практически с любого ЦКП, что улучшает гибкость управления.

**Эффективность использования сетевых ресурсов.** В виртуальном режиме КП оптимальный маршрут выбирается только в момент установления логического соединения, поэтому при быстром изменении ситуации на сети путь, оптимальный для первого пакета сообщения, может быть не оптимальным для последующих пакетов одного и того же сообщения.

При дейтаграммном режиме коррекция маршрута производится чаще, что позволяет более равномерно загрузить каналы всей сети и, в конечном счёте, уменьшить время доставки сообщения.

## **2. Сфера применения архитектур ISO и TCP/IP**

Сфера применения архитектур ISO и TCP/IP определяется их свойствами, которые порождают основные достоинства и недостатки используемых сетевых архитектур.

Так, к основным достоинствам архитектуры ISO следует отнести:

- возможность реализации сетей даже на плохих каналах связи за счёт развитой системы защиты от ошибок и сбоев;
- возможность работать в реальном масштабе времени, простота реализации режима диалога и передачи речи в цифровой форме, поскольку задержки в доставке пакетов одного и того же сообщения незначительны;
- высокая степень стандартизации протоколов на всех уровнях, что упрощает построение ИВС заданных размеров с требуемыми показателями качества обслуживания.

Недостатки архитектуры ISO следующие:

- высокая избыточность за счёт большого объёма необходимой служебной информации;

- необходимость реализации большого набора достаточно сложных протоколов взаимодействия, причём отсутствие хотя бы одного протокола приводит к невозможности передачи данных;

- существенные трудности при организации взаимодействия различных сетей, особенно при различной сетевой архитектуре.

Рассмотрим теперь основные достоинства и недостатки архитектуры ТСП/IP.

Достоинства архитектуры ТСП/IP:

- небольшие затраты на реализацию протоколов взаимодействия за счёт меньшего набора требуемых протоколов;

- существенное упрощение процедуры маршрутизации, что снижает стоимость базовой сети передачи данных за счёт использования более простых ЦКП;

- возможность построения крупномасштабной ИВС с использованием разнотипного оборудования;

- возможность реализации взаимодействия различных сетей с применением простых алгоритмов согласования.

К недостаткам архитектуры ТСП/IP можно отнести:

- возможность реализации только при использовании «хороших» каналов связи (желательно выделенных);

- необходимость решения проблемы сборки пакетов, которые могут поступать на транспортный уровень в произвольном порядке;

- возможность потери сообщения из-за несвоевременной доставки одного из пакетов этого сообщения;

- усложнение прикладных программ пользователя за счёт введения процедур контроля и исправления ошибок в получаемых сообщениях.

Теперь, опираясь на проведённый анализ, можно определить сферу применения сетевых архитектур.

Сетевая архитектура ISO эффективна при применении «плохих» каналов связи, необходимости работы в реальном масштабе времени и

однородной структуре оборудования, причём основным выступает качество каналов связи.

При построении глобальных сетей, когда решающим фактором выступает простота согласования работы различных национальных сетей, реализуемых, как правило, на разнотипном оборудовании, наиболее эффективно применение архитектуры TCP/IP, данный вывод подтверждается практикой, т.к. в Internet используют именно архитектуру TCP/IP.

### **Контрольные вопросы:**

1. Каковы основные требования к сетевым архитектурам?
2. Проведите сравнение сетевых архитектур по стекам протоколов.
3. Проведите сравнение архитектур ISO и TCP/IP по идеологии защиты от ошибок.
4. Проведите сравнение виртуального и дейтаграммного режимов КП.
5. Дайте анализ достоинств и недостатков сетевых архитектур ISO и TCP/IP.

## **1. Прикладной уровень. Протоколы FTP, Telnet, SMTP, POP, HTTP**

### **1.1. Служба FTP. Протокол FTP. Протокол TFTP**

#### **Служба FTP**

Служба FTP предназначена для обмена файлами и построена по технологии "клиент-сервер".

Взаимодействие клиента и сервера осуществляется по протоколу FTP (File Transfer Protocol – протокол передачи файлов).

**Клиент** посылает запросы серверу, принимает и передает файлы.

**Сервер** обрабатывает запросы клиента, передает и принимает файлы.

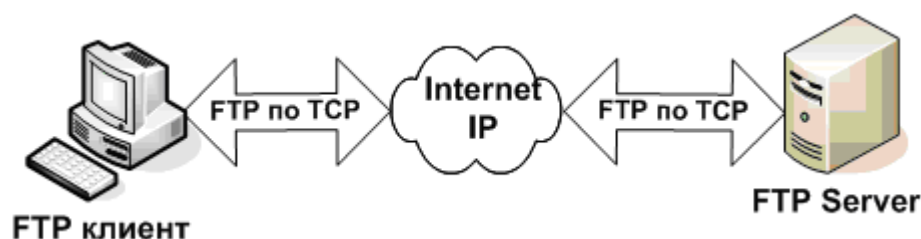


Рис. 1.1. Взаимодействие клиента и сервера по протоколу FTP

FTP-клиент – это программный интерфейс пользователя, реализующий протокол передачи файлов FTP.

Эта программа позволяет пользователю передавать файлы между двумя компьютерами, связанными между собой локальной (LAN) или глобальной (WAN) сетью. При этом компьютерные платформы могут быть различных типов.

FTP-серверы, как правило, доступны только для зарегистрированных пользователей и требуют при подключении ввода идентификатора (login – входное имя) и пароля (password).

Многие FTP-серверы открыты и для свободного доступа, их часто называют анонимными.

Для таких серверов login (входное имя) – anonymous, а в качестве пароля (password) рекомендуют ввести адрес своей электронной почты.

Большинство Web-браузеров обеспечивают доступ к FTP-серверам без использования специальных FTP-клиентов.

Например, URL-адрес:

`ftp://ftp.ware.ru/pub/win/internet/ftp/dl.zip`

означает “связаться с FTP-сервером с правами для анонимных пользователей, перейти в каталог pub, далее в каталог win, каталог internet, каталог ftp и взять файл dl.zip”.

Сеанс работы с FTP-сервером можно провести в режиме командной строки. Для этого необходимо ввести команду `ftp` и после пробела ввести IP-адрес или DNS-адрес FTP-сервера. Если регистрация прошла успешно и связь установлена, то с помощью команд FTP можно выполнить все действия по работе с файлами.

### **Протокол FTP**

Протокол FTP (File Transfer Protocol – протокол передачи файлов) используется службой FTP для передачи файлов, непосредственно взаимодействует с протоколом транспортного уровня TCP.

Первый стандарт – **RFC-114** (File Transfer Protocol A.K. Bhushan Apr-10-1971).

Последняя версия – **RFC-959** (File Transfer Protocol J. Postel, J.K. Reynolds Oct-01-1985).

FTP отличается от других приложений тем, что он использует два TCP соединения для передачи файла:

1. **Управляющее соединение** – соединение для отправки команд серверу и получения ответов от него.

Для организации такого соединения используется протокол Telnet. Telnet-соединение устанавливается в один шаг – отправка запроса и ожидание ответа, получение которого свидетельствует о возможности передачи команд FTP.

Канал управления существует на протяжении всей FTP-сессии и закрывается после завершения информационного обмена.

2. **Соединение данных** – соединение для передачи файлов.

Передача файлов после установленного Telnet-соединения осуществляется через логическое соединение, организуемое протоколом TCP, который проверяет доступность портов, закрепленных за FTP.

Канал данных формируется и ликвидируется по мере необходимости.

Протокол FTP предусматривает два возможных режима установления связи для обмена файлами:

- активный режим;
- пассивный режим.

#### Активный режим

Действия клиента и сервера:

1. Клиент устанавливает связь и посылает с нестандартного порта N (  $N > 1024$  ) запрос на порт сервера;
2. Сервер посылает ответ на порт N клиента;
3. Сервер устанавливает связь для передачи данных по порту 20 на порт клиента N+1.

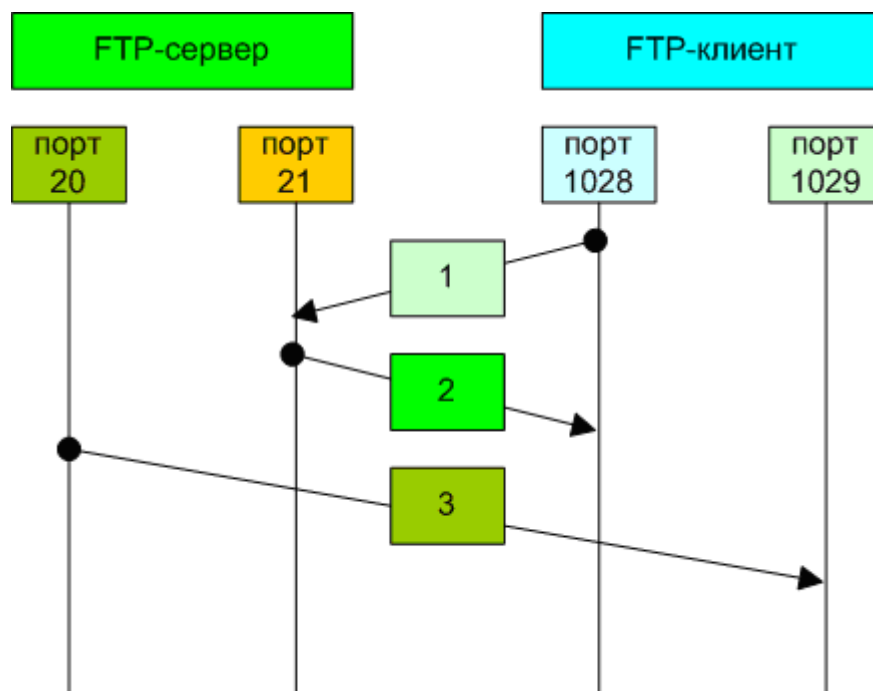


Рис. 1.2. [Пример установления связи для обмена файлами в активном режиме](#)

**Активный режим** выгоден для FTP-сервера, но вреден для клиента. Так как FTP сервер пытается соединиться со случайным высоким (по номеру) портом на клиенте, то такое соединение может быть заблокировано брандмауэром на стороне клиента.

#### Пассивный режим

Действия клиента и сервера:

1. Клиент устанавливает связь и посылает запрос (сообщает, что надо работать в пассивном режиме) на 21 порт сервера с нестандартного порта  $N$  ( $N > 1024$ );
2. Сервер назначает нестандартный порт  $P$  для канала данных ( $P > 1024$ ) и посылает на порт  $N$  клиента ответ, в котором сообщает номер порта  $P$ ;
3. Клиент устанавливает связь для передачи данных по порту  $N+1$  на порт сервера  $P$ .

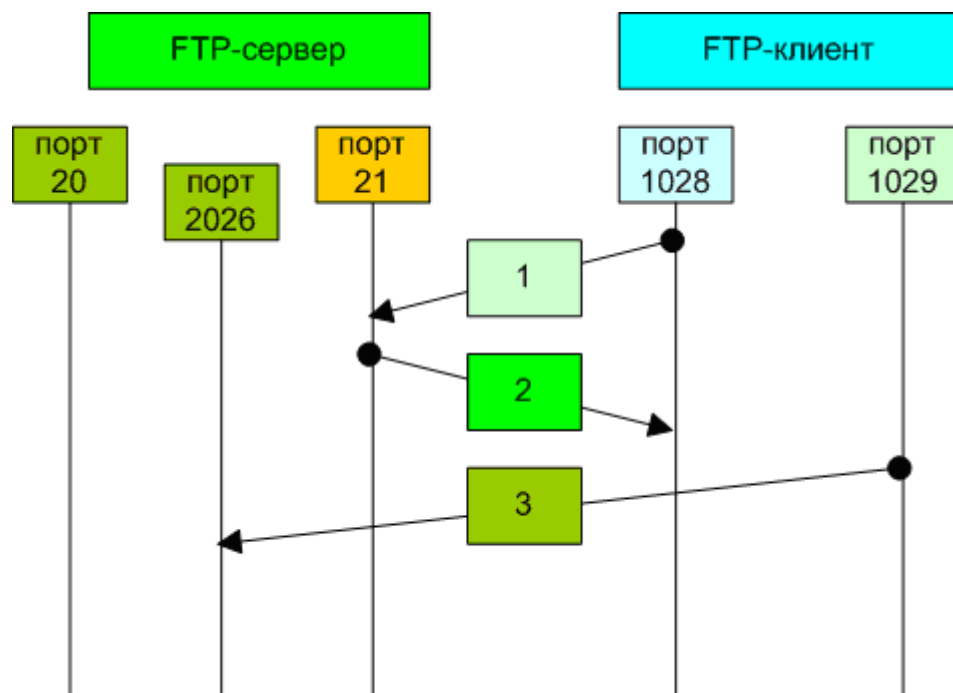


Рис. 1.3. [Пример установления связи для обмена файлами в пассивном режиме](#)

**Пассивный режим** выгоден для клиента, но вреден для FTP-сервера. Клиент будет делать два соединения к серверу, при этом второе будет к случайному высокому порту. Такое соединение может быть заблокировано брандмауэром на стороне сервера.

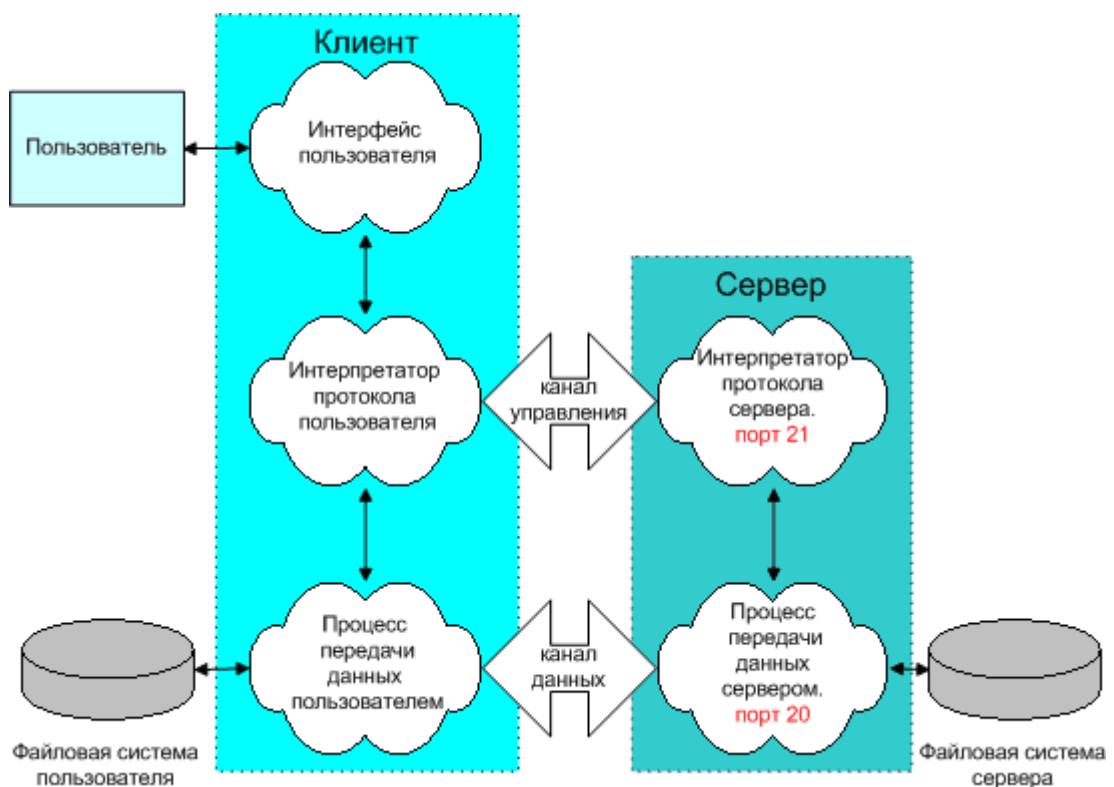


Рис. 1.4. Схема соединения по протоколу FTP в активном режиме

Работа FTP на пользовательском уровне при передаче файлов содержит несколько этапов:

1. Идентификация (ввод имени-идентификатора и пароля);
2. Выбор каталога;
3. Определение режима обмена:
  - передача файлов в текстовом виде;
  - передача файлов в бинарном виде;
4. Выполнение команд обмена;
5. Завершение работы.

[Протокол FTP определяет запрос-ответный способ взаимодействия между программой-клиентом и программой-сервером.](#)

Примеры команд и сеансов работы с FTP-сервером: [прил. 1.1](#), [прил. 1.2](#), [прил. 1.3](#), [прил. 1.4](#), [прил. 1.5](#), [прил. 1.6](#).

### **Протокол TFTP**

Иногда для передачи файлов используется упрощенный протокол TFTP (Trivial File Transfer Protocol – простейший протокол передачи файлов).

Он работает поверх транспортного протокола UDP. Обмен осуществляется в стартстопном режиме (с подтверждением доставки каждого переданного информационного блока) пакетами стандартной длины (512 байт) с заголовком (4 байта). Контроль ошибок не выполняется. Пакет считается принятым, если на него получено подтверждение, в противном случае выполняется однократная повторная передача.

Для начала обмена клиент случайным образом выбирает уникальный идентификатор TID (номер порта) и выдает на сервер запрос на чтение (RRQ) или запись (WRQ) файла. Запрос отправляется на UDP-порт 69 (порт инициализации). Сервер также случайным образом выбирает номер порта (уникальный идентификатор TID) и выдает клиенту подтверждение готовности (ACK). Затем начинается передача файла (команда DATA, блок данных, номер блока, ...). Соединение закрывается после передачи пакета, имеющего длину менее 512 байт, и при получении пакета ERROR.

Команды протокола TFTP:

1. Read request (RRQ) – запрос на чтение файла;
2. Write request (WRQ) – запрос на запись файла;

3. Data (DATA) – пакет данных;
4. Acknowledgment (ACK) – подтверждение;
5. Error (ERROR) – ошибка.

Обычно TFTP используется при загрузке ОС в бездисковые рабочие станции и для загрузки конфигурационных файлов в маршрутизатор.

Отсутствие авторизации в TFTP может явиться одной из угроз безопасности.

```
Microsoft Windows XP [Версия 5.1.2600]  
(C) Корпорация Майкрософт, 1985-2001.  
C:\Documents and Settings\User>tftp ?
```

Обмен файлами с удаленным компьютером, на котором запущена служба TFTP.

TFTP [-i] узел [GET | PUT] источник [адресат]

|          |                                                                                                                  |
|----------|------------------------------------------------------------------------------------------------------------------|
| -i       | Задание режима передачи файлов в двоичном формате.<br>В этом режиме файл передается в неизменном виде по байтам. |
| узел     | Указание локального или удаленного узла.                                                                         |
| GET      | Передача исходного файла с удаленного узла на локальный узел.                                                    |
| PUT      | Передача исходного файла с локального узла на удаленный узел.                                                    |
| источник | Передаваемый файл.                                                                                               |
| адресат  | Задание адресата, которому нужно передать файл.                                                                  |

### 3. Транспортный уровень. Протоколы TCP и UDP

На транспортном уровне используется два основных транспортных протокола UDP — User Datagram Protocol, RFC 768 (протокол пользовательских дейтаграмм) и TCP — Transmission Control Protocol, RFC 793 (протокол управления передачей). Эти протоколы предоставляют разные услуги прикладным процессам, причём большинство хостов активно используют только один из них.

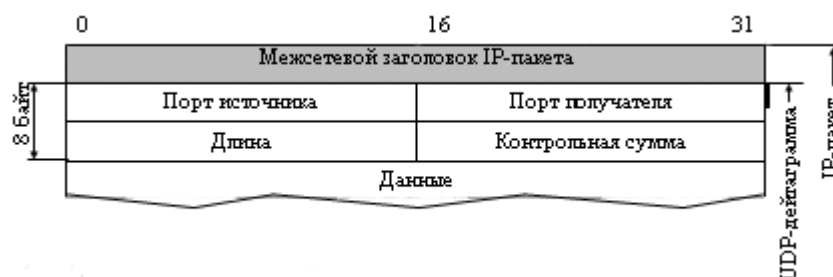
Если требуется надёжная и эффективная доставка сообщений по протяженному и ненадёжному каналу ПД, то применяется протокол TCP. Если требуется передавать сообщения на высокоскоростных сетях с короткими соединениями, то лучше подходит протокол UDP.

#### Протокол UDP

Данный протокол иногда называют протоколом ненадёжной доставки. Этот протокол предоставляет прикладным процессам транспортные услуги, которые немногим отличаются от услуг протокола IP (сетевого уровня).

Протокол UDP обеспечивает только доставку дейтаграммы и не гарантирует её выполнение. При обнаружении ошибки дейтаграмма просто стирается. Протокол не поддерживает виртуального соединения с удалённым модулем UDP. Чаще всего базируется на принципах динамической маршрутизации (каждая дейтаграмма передаётся по оптимальному маршруту). Основное достоинство — простота.

Формат UDP-дейтаграммы имеет следующий вид:



Видно, что формат протокола UDP размещается в поле данных IP-пакета (или после заголовка IP-пакета) и содержит следующие поля:

Поле **«Порт источника»** (Source Port) указывает порт процесса источника, куда может быть адресован ответ на данное сообщение.

Поле **«Порт получателя»** (Destination Port) идентифицирует принимающий процесс. Под «портом» понимается адрес (номер) некой точки доступа к услугам другого уровня. В случае архитектуры ТСП/IP под портом понимается некий номер области памяти, где размещаются передаваемые в сеть (протоколу UDP или TCP) и принимаемые из сети (поступающие в распоряжение операционной системы) данные. Номера портов на передачу и приём в общем случае могут различаться. На приёмной и передающей сторонах взаимодействие процессов в общем случае может происходить через разные номера портов, поэтому указание порта в заголовке UDP-дейтаграммы необходимо.

В поле **«Длина»** (Length) указывается размер данной дейтаграммы с учётом длины заголовка в байтах.

Поле **«Контрольная сумма»** (Checksum) обеспечивает контроль правильности данных и заголовка. Суммируются все контролируемые 16-

битные слова (с циклическим переносом из старшего разряда в младший). Инвертированное значение результата записывается в поле контрольной суммы. Если UDP-дейтаграмма содержит нечетное число байтов, то недостающий последний байт в таких случаях считается нулевым. Этот байт не передается в области данных.

При подсчете контрольной суммы протокол UDP учитывает 12-байтовый [псевдозаголовок](#) (*pseudo header*). Псевдозаголовок включает в себя: IP-адрес источника, IP-адрес приемника, протокол (код 17) и длину UDP-дейтаграммы.

Если источник проставил контрольную сумму, а адресат при ее проверке обнаружил ошибку, то UDP-дейтаграмма "молчаливо отбрасывается" - не генерируется никакого сообщения об ошибке.

UDP-данные могут отсутствовать.

Прикладные процессы и модули UDP взаимодействуют через UDP-порты, которые нумеруются, начиная с 0. Прикладной процесс ожидает сообщение в порт, специально выделенный для этих услуг. Номер этого порта является общеизвестным и определяется стандартами сети Internet.

| Номера портов, используемые в протоколе UDP |                           |                      |                                            |
|---------------------------------------------|---------------------------|----------------------|--------------------------------------------|
| Номер порта                                 | Идентификатор в стандарте | Идентификатор в UNIX | Описание                                   |
| 7                                           | ECHO                      | echo                 | Эхо                                        |
| 15                                          | -                         | netstat              | Программа выдачи состояния сети            |
| 37                                          | TIME                      | time                 | Текущее время                              |
| 42                                          | NAMESERVER                | name                 | Сервер имен хостов                         |
| 43                                          | NICNAME                   | whois                | Информация о пользователе                  |
| 53                                          | DOMAIN                    | nameserver           | Сервер доменных имен (DNS)                 |
| 67                                          | BOOTPS                    | bootps               | Сервер BOOTP или DHCP                      |
| 68                                          | BOOTPC                    | bootpc               | Клиент BOOTP или DHCP                      |
| 69                                          | TFTP                      | tftp                 | Простейший протокол передачи файлов (TFTP) |
| 88                                          | KERBEROS                  | kerberos             | Служба аутентификации Kerberos             |
| 123                                         | NTP                       | ntp                  | Протокол синхронизации сетевого            |

|     |   |           |                                             |
|-----|---|-----------|---------------------------------------------|
|     |   |           | времени (NTP)                               |
| 161 | - | snmp      | Простой протокол сетевого управления (SNMP) |
| 162 | - | snmp-trap | Прерывания протокола SNMP                   |
| 512 | - | biff      | Сервер UNIX comsat                          |

Более подробную информацию о протоколе UDP можно найти в **RFC-768**

### Протокол TCP

Данный протокол является сквозным и ориентирован на создание соединений, то есть в данном протоколе создаётся виртуальное соединение. Он работает в широком спектре систем связи — от выделенных каналов до сетей с КП или КК. Протокол TCP размещается над сетевым протоколом IP, который даёт возможность TCP посылать и принимать сегменты информации различной длины, вложенные в межсетевые дейтаграммные «конверты» (пакеты).

При организации связи между парой прикладных процессов протокол TCP обеспечивает следующее: надёжную передачу данных; управление потоком данных; мультиплексирование; организацию, поддержание и сброс виртуального соединения (виртуального канала); приоритетную доставку информации и её безопасность.

Протокол TCP, в отличие от протокола UDP, создаёт виртуальные соединения или каналы. Подобно модулю UDP, прикладные процессы взаимодействуют с модулем TCP через порты, которые имеют общеизвестные адреса (номера).

| Часто используемые стандартные номера портов протокола TCP |                          |                      |                                        |
|------------------------------------------------------------|--------------------------|----------------------|----------------------------------------|
| Номер порта                                                | Идентификатор в Internet | Идентификатор в UNIX | Описание                               |
| 7                                                          | ECHO                     | echo                 | Эхо                                    |
| 20                                                         | FTP-DATA                 | ftp-data             | Данные протокола передачи файлов (FTP) |
| 21                                                         | FTP                      | ftp                  | Протокол передачи файлов (FTP)         |

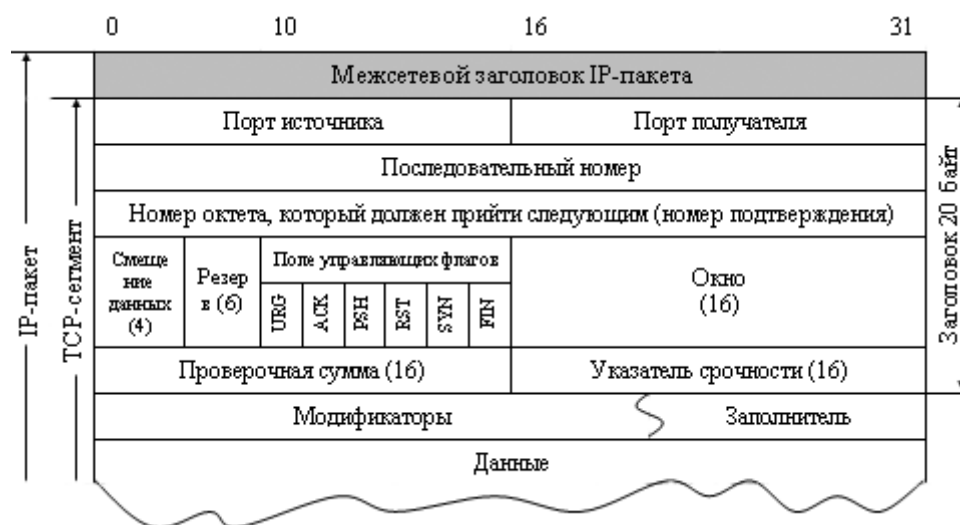
|     |           |            |                                                    |
|-----|-----------|------------|----------------------------------------------------|
| 23  | TELNET    | telnet     | Подключение к терминалу                            |
| 25  | SMTP      | smtp       | Простой протокол передачи электронной почты (SMTP) |
| 37  | TIME      | time       | Текущее время                                      |
| 43  | NICNAME   | whois      | Информация о пользователе                          |
| 53  | DOMAIN    | nameserver | Сервер доменных имен (DNS)                         |
| 67  | BOOTPS    | bootps     | Сервер BOOTP или DHCP                              |
| 80  | WWW       | www        | Сервер World Wide Web                              |
| 88  | KERBEROS  | kerberos   | Служба аутентификации Kerberos                     |
| 101 | HOSTNAME  | hostname   | Сервер NIC имен хостов                             |
| 103 | X400      | x400       | Почтовая служба X.400                              |
| 104 | X400-SND  | x400-snd   | Отправитель почты X.400                            |
| 113 | AUTH      | auth       | Служба аутентификации                              |
| 117 | UUCP-PATH | uucp-path  | Служба путей протокола UUCP                        |
| 119 | NNTP      | nntp       | Протокол передачи сетевых новостей Usenet (NNTP)   |
| 123 | NTP       | ntp        | Протокол синхронизации сетевого времени (NTP)      |
| 161 | SNMP      | snmp       | Простой протокол сетевого управления (SNMP)        |

Когда прикладной процесс начинает использовать ТСР, то этот модуль на хосте и модуль ТСР на сервере приложений начинают взаимодействовать. Эти два оконечных модуля, прежде всего, создают виртуальное соединение, которое является дуплексным и расходует ресурсы обоих оконечных модулей ТСР. Протокол ТСР разбивает поток двоичных разрядов (поступающих с вышележащего уровня) на ТСР-сегменты, которые передаются по виртуальному соединению. На приёмном конце производится обратная операция.

Протокол ТСР требует, чтобы все отправленные сегменты данных были подтверждены с приёмного конца, т.е. используется алгоритм обратной связи. Для повышения эффективности работы используются

механизм скользящего окна, тайм-ауты и повторные передачи для обеспечения надёжной доставки. Каждая из принимающих сторон может управлять потоком данных от передающего модуля, чем предотвращается возможность переполнения буферов приёмников. Пользователь при установлении соединения может устанавливать категорию срочности и безопасности. Эти признаки учитываются не только при работе с ТСР-сегментами, но и дублируются в поле «Тип сервиса» IP-пакета.

Формат ТСР-сегмента включает [заголовок](#) и данные и имеет следующий вид:



Поля «**Порт источника**» и «**Порт получателя**» указывают номера портов в ТСР-модулях (идентифицируют взаимодействующие приложения).

В поле «**Последовательный номер (Позиционный номер)**» содержится номер, который указывает место в потоке данных от источника до конечного получателя первого байта содержащихся в этом сегменте данных.

В начальном сегменте, посылаемом при установлении соединения, присутствует флаг SYN, а в поле «Позиционный номер» содержится так называемый *начальный позиционный номер ISN (initial sequence number)*, выбранный данным хостом для этого нового соединения. Первому байту данных, переданному хостом по новому соединению, будет присвоен позиционный номер, равный ISN+1.

Поле «**Номер подтверждения (Квитанция)**» при установленном флаге ACK содержит значение последовательного номера, который отправитель данного сегмента собирается принимать. (То есть этот номер всегда на единицу больше номера последнего успешно принятого байта). После установления виртуального соединения это поле обязательно

заполняется. С помощью этого поля отмечается байт, с которого начнется окно приёма данных от источника (механизм скользящего окна).

TCP предоставляет приложениям сервис *полнодуплексной передачи*. Это означает, что данные могут передаваться в обоих направлениях независимыми потоками. В процессе двусторонней передачи каждая из сторон должна вести учет позиционных номеров передаваемых и принимаемых ею байтов.

Поле **«Смещение данных (Размер заголовка)»** определяет число 32-разрядных слов в заголовке TCP-сегмента, так как такое же смещение есть в межсетевом заголовке.

В поле **«Резерв»** все разряды устанавливаются равными 0.

Содержимое **«Поля управляющих флагов»**: URG (urgency) - указатель срочности; ACK (acknowledgement) - подтверждение; PSH (push) - указатель немедленной выдачи на верхний уровень; RST (reset) - немедленный сброс соединения; SYN (synchronization) - синхронизация последовательных номеров; FIN (final) - завершение соединения.

Поле **«Окно»** содержит число байт, равное длине окна, т.е. период времени в байтах, когда отправитель ожидает информацию от приёмника.

Поле **«Проверочная сумма»**. Проверочная сумма подсчитывается для всего TCP сегмента, при этом определяется 16-битное дополнение суммы всех 16-битных слов в заголовке и в поле данных. Если сегмент содержит нечетное количество байтов, то он будет дополнен нулями справа до образования 16-битного слова. Этот выравнивающий байт не передается с сегментом по сети, так как может быть "восстановлен" получателем.

Проверочная сумма учитывает также 96-битный псевдозаголовок, который ставится перед заголовком протокола TCP. Подзаголовок включает следующие поля из заголовка протокола IP: IP-адреса отправителя и получателя, протокол (код 6) и длину сегмента.

С помощью добавления псевдозаголовка протокол TCP защищает самого себя от ошибочной доставки протоколов IP. Так, если протокол IP доставляет сегмент, не предназначенный данному работающему приложению, то модуль протокола TCP на принимающей стороне обнаружит некорректность доставки.

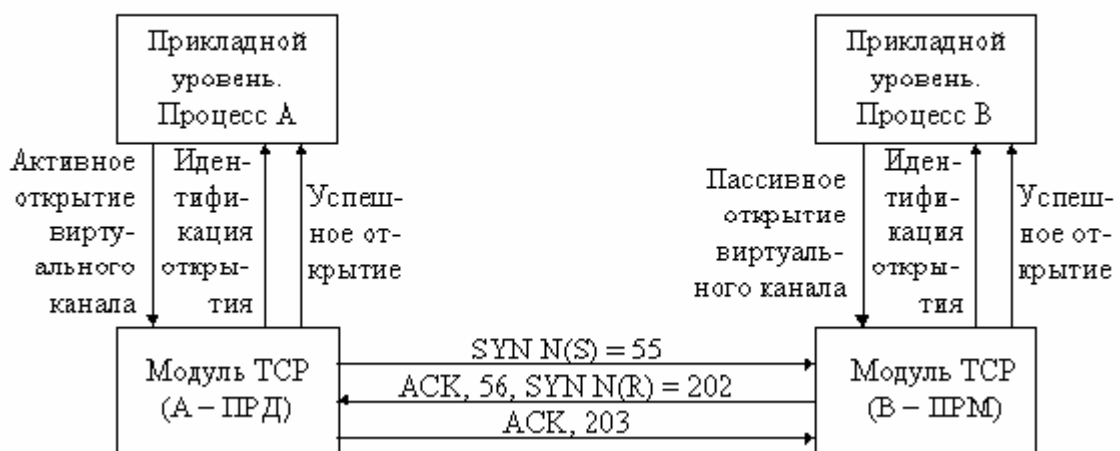
Поле **«Указатель срочности (Указатель границы срочных данных)»** указывает последовательный номер байта, которым заканчиваются срочные данные. Если установлен флаг URG и это поле равно 0, то весь сегмент считается срочным.

Поле «**Модификаторы (Опции)**» указывает на дополнительные услуги и может иметь переменную длину, кратную байту.

Поле «**Заполнитель**» имеет переменную длину и дополняет заголовок до целого числа 32-разрядных слов. Поле заполняется нулями.

Рассмотрим типовой диалог между двумя объектами прикладного уровня с использованием протокола TCP.

*Фаза 1 — установление соединения.*



### **Взаимодействие узлов в режиме «прикладной уровень – TCP»:**

N(S) - номер байта, с которого начнёт передавать передатчик (ПРД), например, 55;

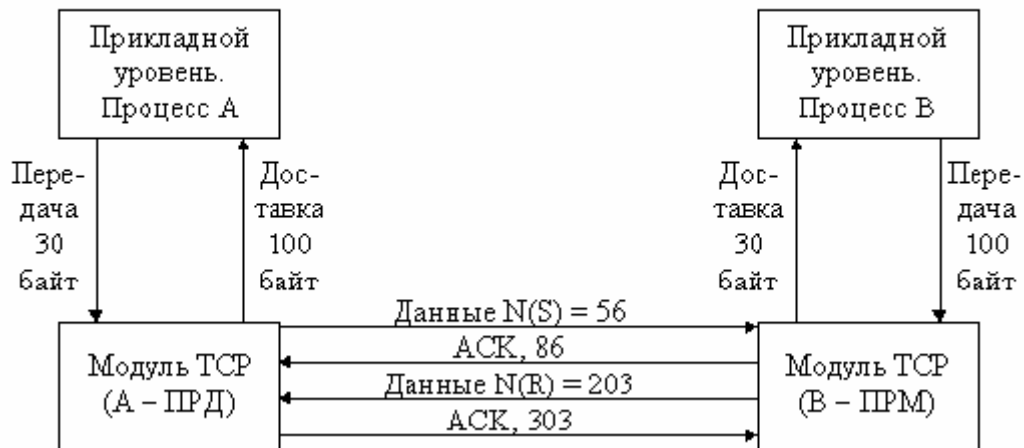
N(R) - номер байта, с которого будет передавать приёмник (ПРМ), например, 202.

Для открытия виртуального соединения посылается флаг SYN в сегменте, с которого начнется передача (N(S)=55). Приёмник отвечает сегменту, в котором флаг ACK установлен в 1 и указывает номер байта, с которого он начнёт передавать (N(R)=202). В заголовке этого же сегмента в поле «Номер подтверждения» приёмник указывает, что он ожидает от передатчика байт с номером 56. Здесь же передаётся флаг синхронизации SYN. Передатчик (модуль А), получив этот сегмент с подтверждением о готовности приёмника работать, также отвечает сегментом с

подтверждением АСК, и в поле «Номер подтверждения» передатчик указывает, что он ожидает от приёмника байт с номером 203.

После этого виртуальное соединение установлено, о чем модули ТСП извещают свои прикладные процессы.

*Фаза 2 — передача данных.*



### ***Взаимодействие узлов в режиме «прикладной уровень – ТСП»***

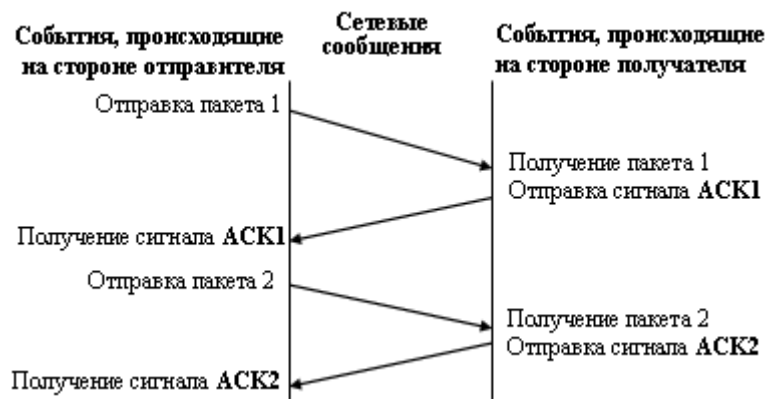
Передатчик по созданному виртуальному соединению передаёт данные (30 байт), начиная с байта под номером 56. Приёмник ожидает байт данных именно с этим номером, поэтому после приёма данных приёмник выдаёт сегмент с флагом подтверждения АСК и номером следующего ожидаемого байта  $N(S)=56+30=86$ , кроме того, приёмник отсылает в сторону передатчика 100 байт данных, начиная с номера 203, что и ожидает передатчик. Получив 100 байт от приёмника, передатчик выдаёт сегмент с флагом АСК и номером следующего ожидаемого байта  $N(R)=203+100=303$ .

В фазе "передача данных" работают механизмы обеспечения надёжной доставки.

Различают три варианта обратной связи:

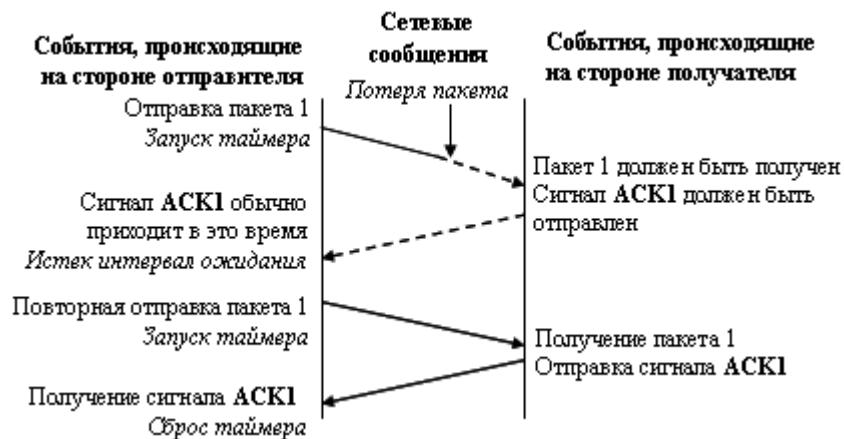
- РОС ОЖ
- Оконный режим переспроса
- Адресный режим переспроса

1. **РОС ОЖ.** Это наиболее простой вариант.



***Иллюстрация механизма подтверждения приема с повторной передачей, при котором отправитель ждет уведомления об успешном получении каждого пакета***

На каждый передающийся сегмент ожидается получение квитанции (ACK=1 и номер следующего запрашиваемого октета). Включается таймер ожидания. Если квитанция не приходит до истечения таймера, то осуществляется повторная передача.



***При потере пакета через определенный интервал времени выполняется его повторная передача. Пунктирной линией показан процесс нормальной передачи пакета и получения подтверждения***

Причинами потери пакетов могут являться ошибки, возникающие в заголовке IP, истечение времени жизни, переполнение буфера маршрутизатора и т.п. Во всех этих случаях отправляется сообщение ICMP. Время ожидания квитанции зависит от расстояния до получателя (от времени двойного пробега).

Существует 2 механизма определения времени двойного пробега:

1. Тестирование сети и оценка в режиме «пингования» временной задержки на каждом сегменте. Затем осуществляется набор статистики и её обработка. Для определения времени ожидания квитанции к среднему времени задержки добавляются 4 среднеквадратических отклонения времени задержки от его математического ожидания. Именно за это время и должна придти квитанция. Если квитанция приходит после истечения таймера, передающая сторона считает пакет утерянным.
2. Динамический способ определения таймера. Время двойного пробега определяется в процессе передачи. Первоначально передаются сегменты по одному в режиме РОС ОЖ и определяется время пробега. Время таймера увеличивается до тех пор, пока квитанции не станут не успевать приходить.

К плюсам РОС ОЖ можно отнести простоту реализации. К минусам - непроизводительное использование пропускной способности и, следовательно, - низкую эффективную скорость ПД.

2. **Оконный режим переспроса.** Для данного режима характерно некоторое число сегментов, передаваемых непрерывно без ожидания квитанции. Оконный режим использует принцип конвейерной передачи. Таймер включается после передачи последнего сегмента.

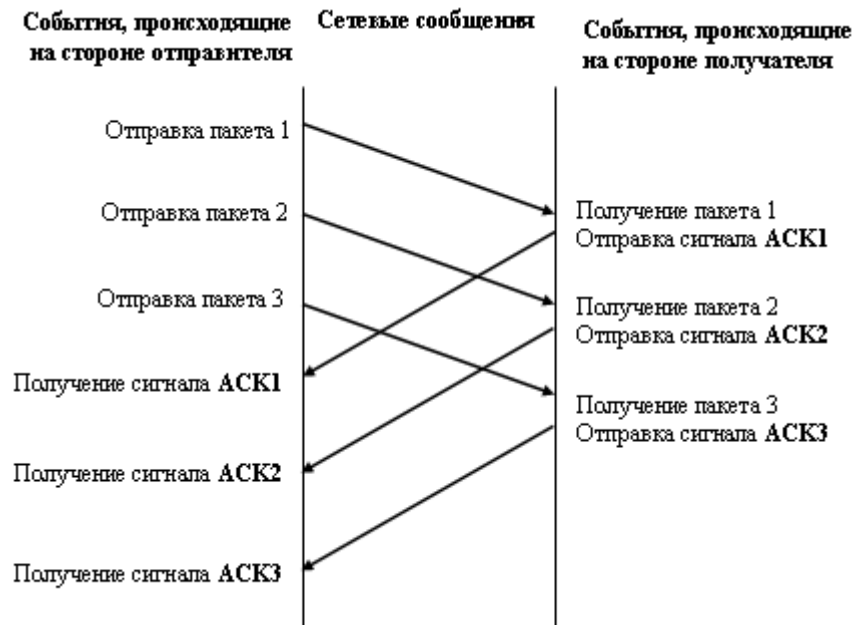


### **Пример движущегося окна протокола TCP**

Октеты 1 и 2 успешно доставлены получателю; октеты 3-6 посланы в сеть, но подтверждение об их доставке еще не получено; октеты 7-9 еще не отправлены, но могут быть отправлены без всяких задержек; октеты с номерами 10 и выше не могут быть посланы в сеть до тех пор, пока не попадут внутрь окна.

Возможны 2 режима окна: фиксированное и скользящее.

- **Режим фиксированного окна.** Переспрос всего окна осуществляется в случае, если хотя бы один сегмент принят с ошибкой (РОС ОЖ, но для всего окна). Эффективность использования лучше, чем при РОС ОЖ.

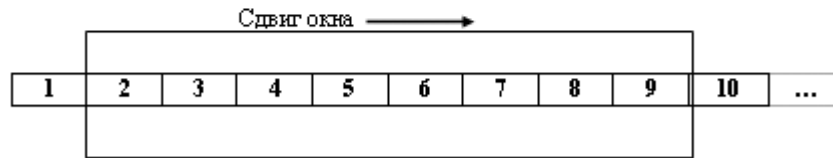


*Пересылка трех пакетов с использованием метода движущихся окон. Идея заключается в том, что отправитель может послать в сеть сразу все три пакета, не дожидаясь сообщений о подтверждении их приема*

- **Режим скользящего окна.** Переспрос осуществляется с первого ошибочного сегмента. В этом режиме подтверждения, следующие за ошибочным сегментом, не передаются. Окно смещается на число правильно принятых сегментов. Этот режим наиболее эффективен.
3. **Адресный переспрос.** Конвеерная передача в размере окна. Все принимаемые сегменты накапливаются и анализируются. Определяются номера сегментов, которые приняты с ошибкой, и осуществляется повторная передача только этих сегментов.



(а)

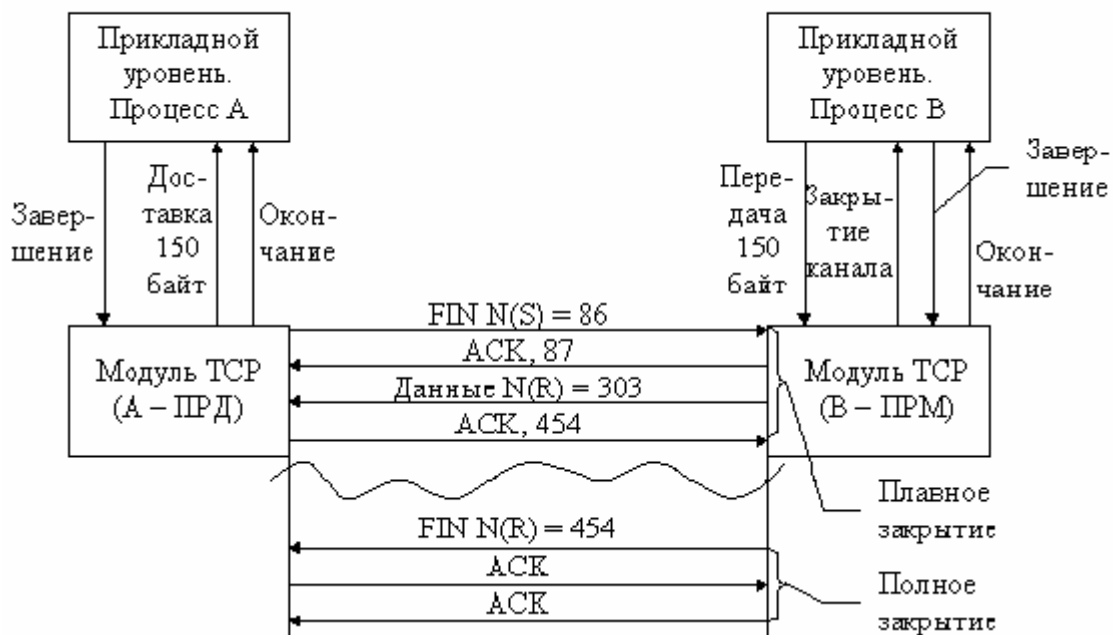


(б)

*Движущееся окно, внутрь которого помещено 8 пакетов (а); при получении подтверждения о приеме пакета 1 окно сдвигается на один пакет вправо, и в сеть отправляется 9-й пакет (б). Если для какого-либо пакета, попавшего в окно, не получен сигнал подтверждения, то выполняется повторная передача этого пакета*

Режим окна определяется во вспомогательных параметрах ТСП заголовка.

*Фаза 3 — Плавное закрытие соединения.*



*Взаимодействие узлов в режиме «прикладной уровень – ТСП»*

Для плавного завершения соединения передатчик отправляет сегмент с флагом FIN и номером байта  $N(S)=86$ . Приёмник выдаёт сегмент с флагом подтверждения ACK и номером ожидаемого байта  $N(S)=87$ , но у приёмника ещё остались данные для передачи, которые он и отправляет (150 байт), начиная с байта под номером  $N(R)=303$ . Передатчик отвечает сегментом с флагом подтверждения ACK и номером ожидаемого байта 454 ( $303+150+1$ ).

На этом виртуальное соединение прикладного уровня разрывается, но остается ещё виртуальное соединение транспортного уровня. Для его разрушения приёмник посылает сегмент с флагом FIN и номером ожидаемого байта  $N(R)=454$ . Передатчик отвечает подтверждением, на что приёмник также отвечает сегментом подтверждения ACK. На этом виртуальное соединение на транспортном уровне разрушается.

Более подробное описание [протокола TCP](#) можно найти в **RFC-793**, **RFC-1180**

*Пример вычисления контрольной суммы (TCP checksum) TCP-сегмента*

#### **Контрольные вопросы:**

1. Опишите услуги протокола UDP.
2. Назначение транспортного протокола TCP.
3. Опишите назначение полей формата TCP-сегмента.
4. Объясните взаимодействие объектов прикладного уровня с помощью TCP.

### **3. Протокол межсетевого взаимодействия (IP). Адресация в IP-сетях**

**Физический и канальный уровень (Уровень сетевых интерфейсов).** стек TCP/IP не подразумевает использования каких-либо определенных протоколов уровня доступа к среде передачи и физических сред передачи данных. От уровня доступа к среде передачи требуется наличие интерфейса с модулем IP, обеспечивающего передачу IP-пакетов. Также требуется обеспечить преобразование IP-адреса узла сети, на который передается IP-пакет, в MAC-адрес. Часто в качестве уровня

доступа к среде передачи могут выступать целые протокольные стеки, тогда говорят об IP поверх ATM, IP поверх IPX, IP поверх X.25 и т.п.

### **3.1. Межсетевой уровень и протокол IP (Internet Protocol)**

Основу межсетевого уровня (уровня межсетевого взаимодействия) составляет IP-протокол.

Первый стандарт IPv4 определен в RFC-760 (1980 г.).

Последняя версия IPv4 – [RFC-791](#) (1981 г.).

Первый стандарт IPv6 определен в RFC-1883 (1995 г.).

Последняя версия IPv6 – [RFC-2460](#) (Internet Protocol, Version 6 Specification, S. Deering, R. Hinden, December 1998).

Пакет, передаваемый по сети Internet, называют **IP-дейтаграммой** или **IP-пакетом**.

Межсетевой протокол IP является дейтаграммным протоколом без установления соединения, не обеспечивающим надежную доставку пакетов.

Основными функциями протокола IP являются:

- перенос между сетями различных типов адресной информации в унифицированной форме;
- сборка и разборка пакетов при передаче их между сетями с различным максимальным значением длины пакета.

Основными характеристиками протокола IP являются:

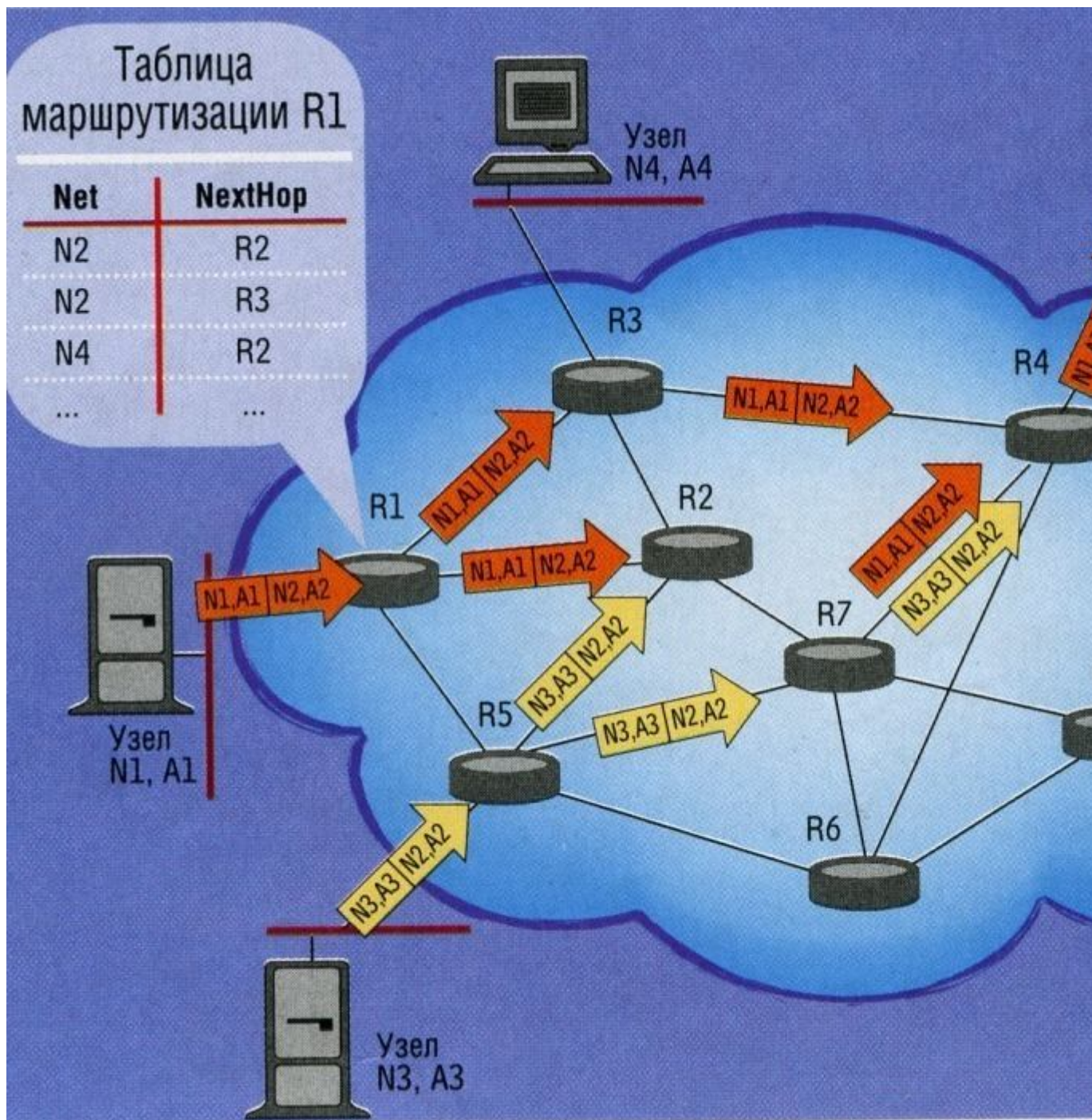
- формат IP пакета;
- способ обработки конфликтных ситуаций;
- способ маршрутизации.

"Ненадежность доставки":

- не гарантируется доставка пакетов получателю;
- по пути следования пакет может быть утерян, продублирован, задержан;

- пакеты могут быть доставлены с нарушением порядка следования.

Для доставки пакетов не требуется предварительного установления соединения (т.е. пути следования пакетов), так как каждый пакет считается независимым от остальных. Поэтому пакеты от отправителя до получателя могут проходить по разным маршрутам.



В протоколе IP применяют четыре основных механизма для обеспечения межсетевых услуг: вид обслуживания, время жизни, контрольная сумма заголовка, дополнительные возможности (опции).

**Вид обслуживания** используется для указания требуемого качества обслуживания межсетевой дейтаграммы (МД) при её передаче через межсетевую систему.

**Время жизни** является указателем верхней границы времени существования некоторой межсетевой дейтаграммы в сети. Этот указатель задаётся отправителем и уменьшается по мере движения МД по точкам маршрута (по шлюзам), если время МД становится нулевым до того, как она достигнет получателя, то эта дейтаграмма уничтожается.

**Контрольная сумма заголовка** обеспечивает защиту данных в нём, если модуль обнаруживает ошибку в заголовке, то эта МД уничтожается модулем, который её обнаружит.

**Дополнительные возможности** обеспечивают выполнение некоторых дополнительных услуг, например, по защите данных, специальной маршрутизации.

#### Межсетевая дейтаграмма

##### Структура IP-пакета: заголовок и блок данных.

В заголовок IP-пакета включен **набор правил**, обеспечивающих доставку пакета данных получателю. В этих правилах оговариваются способы обработки пакетов узлами сети и маршрутизаторами, а также условия, при возникновении которых должны генерироваться сообщения об ошибке, а пакеты удаляться из сети.

|                                         |                                          |                                     |                                       |                                           |
|-----------------------------------------|------------------------------------------|-------------------------------------|---------------------------------------|-------------------------------------------|
| 4                                       | 4                                        | 8                                   | 16                                    |                                           |
| Версия<br>(Version)                     | Длина<br>заголовка<br>(Header<br>Length) | Тип сервиса<br>(Type of<br>Service) | Полная длина пакета<br>(Total Length) |                                           |
| 16                                      |                                          |                                     | 3                                     | 13                                        |
| Общий идентификатор<br>(Identification) |                                          |                                     | Флаг<br>(Flag)                        | Фрагментное смещение<br>(Fragment Offset) |

|                                                      |                                |                                                       |
|------------------------------------------------------|--------------------------------|-------------------------------------------------------|
| 8                                                    | 8                              | 16                                                    |
| Время жизни<br>(TTL - Time To Live)                  | Тип<br>протокола<br>(Protocol) | Контрольная сумма заголовка<br>(Header Checksum)      |
| IP-адрес отправителя<br>(Source Address)             |                                |                                                       |
| IP-адрес получателя<br>(Destination Address)         |                                |                                                       |
| Вспомогательные параметры IP (опции IP)<br>(Options) |                                | Заполнитель<br>(Padding)<br>(дополнение до 32<br>бит) |
| Данные<br>(Data)                                     |                                |                                                       |
| ...                                                  |                                |                                                       |

Рис. 3.1. Формат IP-пакета

Функциональное назначение полей заголовка.

Поле **Версия (Version)** указывает номер версии данного протокола межсетевого уровня. В настоящее время наряду с 4-й версией протокола (т.е. в поле — 0100) начинается использование протокола 6-й версии (т.е. в поле — 0110).

Поле **Длина заголовка (Header Length)** указывает длину заголовка межсетевой дейтаграммы в 32-разрядных словах. Минимальная длина — пять слов, максимальная длина — пятнадцать 32-разрядных слов (на рисунке заголовок имеет шесть слов).

Поле **Тип сервиса (Type of Service)** указывает параметры требуемого качества обслуживания.

| 9                         | 10 | 11 | 12                 | 13                                   | 14                    | 15                       | 16     |
|---------------------------|----|----|--------------------|--------------------------------------|-----------------------|--------------------------|--------|
|                           |    |    | D                  | T                                    | R                     | C                        |        |
| Приоритет                 |    |    | 1                  | 1                                    | 1                     | 1                        | Резерв |
| 0                         | 0  | 0  | низкая<br>задержка | высокая<br>пропускная<br>способность | высокая<br>надёжность | минимальная<br>стоимость |        |
| обычная информация        |    |    |                    |                                      |                       |                          |        |
| 0                         | 0  | 1  |                    |                                      |                       |                          |        |
| приоритетная информация   |    |    |                    |                                      |                       |                          |        |
| 0                         | 1  | 0  |                    |                                      |                       |                          |        |
| "Срочно"                  |    |    |                    |                                      |                       |                          |        |
| 0                         | 1  | 1  |                    |                                      |                       |                          |        |
| "Молния"                  |    |    |                    |                                      |                       |                          |        |
| 1                         | 0  | 0  |                    |                                      |                       |                          |        |
| "сверхмолния"             |    |    |                    |                                      |                       |                          |        |
| 1                         | 1  | 1  |                    |                                      |                       |                          |        |
| сетевой управляющий пакет |    |    |                    |                                      |                       |                          |        |

Рис. 3.2. Поле Тип сервиса (Type of Service)

Приоритет (*precedence*) предоставляет возможность присвоить код приоритета каждой дейтаграмме.

Биты: 12 - D (delay) — задержка, 13 - T (throughput) — производительность (пропускная способность), 14 - R (reliability) — надёжность, C (cost) — стоимость. Только один бит может быть установлен в 1. По умолчанию все четыре бита равны 0, что означает обычный сервис.

Поле **Полная длина пакета (Total Length)** определяет общую длину дейтаграммы в октетах (байтах), включая заголовок и полезную нагрузку. Полная длина пакета может достигать 65535 байт ( $2^{16} - 1 = 65\,535$ ). Рекомендуется использовать дейтаграмму длиной 576 байт (т.е. 4608 разрядов) — 552 байта данные плюс 24 байта заголовок.

Поле **Общий идентификатор (Identification)** предназначено для сборки фрагментов межсетевых дейтаграмм.

Поле **Флаг (Flag)** обеспечивает возможность фрагментации дейтаграмм и, при использовании фрагментации, позволяет идентифицировать последний фрагмент дейтаграммы.

| 17         | 18                            | 19                         |
|------------|-------------------------------|----------------------------|
| 0 - резерв | 0 - фрагментация разрешена    | 0 - последний фрагмент     |
|            | 1 - фрагментация не разрешена | 1 - промежуточный фрагмент |

<="" p="">

Рис. 3.3. Поле Флаг (Flag)

Поле **Фрагментное смещение** указывает место данного фрагмента в межсетевой дейтаграмме. Первый фрагмент имеет смещение, равное нулю.

Для устранения из сети пакетов, задержанных вследствие каких-либо причин, в заголовке в поле **Время жизни (TTL - Time To Live)** указывается время, в течение которого пакет должен существовать в сети. Значение этого времени уменьшается при прохождении пакета по сети, а по его истечении пакет уничтожается с уведомлением отправителя соответствующим ICMP-сообщением. Такая мера защищает сеть от циклических маршрутов и от перегрузок.

«Время жизни» задаётся в секундах — максимально 255 секунд (приблизительно 4,3 минуты). Однако часто в этом поле указывается максимальное количество хостов, через которые может пройти дейтаграмма. Это является полезным в том случае, когда задержки в сети имеют достаточно большие значения; тогда даже при суммарной задержке более 255 секунд есть вероятность доставки дейтаграммы получателю, если количество транзитных хостов не превысило максимально допустимое значение, определённое в данном поле.

Поле **Тип протокола (Protocol)** идентифицирует протокол верхнего уровня (ICMP - 1, IGMP - 2, TCP - 6, UDP - 17), который будет использован при обработке поля данных межсетевой дейтаграммы.

Протоколы транспортного уровня (протоколы TCP или UDP), пользующиеся сетевым уровнем для отправки пакетов, считают, что максимальный размер поля данных IP-пакета равен 65535, и поэтому могут передать ему сообщение такой длины для транспортировки через сеть. В функции протокола IP входит разбиение слишком длинного для конкретного типа составляющей сети сообщения на более короткие пакеты с созданием соответствующих служебных полей, нужных для последующей сборки фрагментов в исходное сообщение.

В большинстве типов локальных и глобальных сетей определяется такое понятие как максимальный размер поля данных кадра или пакета, в которые должен **инкапсулировать** свой пакет протокол IP (MTU - Maximum Transfer Unit). Так, например, сети Ethernet имеют значение MTU, равное 1500 байт, сети FDDI - 4096 байт, а сети X.25 чаще всего работают с MTU в 128 байт.

Поле **Контрольная сумма заголовка (Header Checksum)**. Для уменьшения вероятности искажения адресной части пакета и, как результат, отправки его не по адресу (и потере) заголовков пакета препроводжается проверочной последовательностью - *контрольной*

*суммой*, занимающей 2 байта и рассчитываемой по всему заголовку.

Для вычисления контрольной суммы IP-заголовка в исходящей дейтаграмме значение этого поля сначала устанавливается в 0. Затем выполняется сложение (с циклическим переносом из старшего разряда в младший) всех 16-разрядных слов заголовка, и инвертированное значение результата записывается в поле контрольной суммы. При получении IP-дейтаграммы вновь вычисляется сумма 16-разрядных слов заголовка. Так как в заголовке принятой дейтаграммы уже содержится сосчитанная (и инвертированная) отправителем контрольная сумма, в результате должно получиться слово, состоящее только из единиц (если в заголовке ничего не изменилось). Если же получилась другая комбинация (ошибка контрольной суммы), IP-модуль уничтожает дейтаграмму. Никакого сообщения об ошибке не порождается. Обнаружение потери дейтаграммы и повторная передача считаются проблемой, решаемой на вышестоящих уровнях иерархии протоколов.

Поскольку некоторые поля заголовка меняются в процессе движения пакета (например, время жизни), то проверочные разряды пересчитываются в каждой точке обработки межсетевой дейтаграммы.

(Алгоритмы подсчета контрольных сумм в протоколах Internet и их реализация описаны в RFC 1071).

#### *Пример вычисления контрольной суммы заголовка (Header Checksum) IP-пакета*

IP-адреса, содержащиеся в заголовке (**IP-адрес отправителя (Source Address)**, **IP-адрес получателя (Destination Address)**), являются 32-битовыми идентификаторами объектов сети - конечных установок и маршрутизаторов. Они состоят из идентификаторов сетей и идентификаторов конечных установок, причем в зависимости от класса сети число бит, выделяемых для идентификации сетей и конечных установок, может меняться. Предусмотрена также возможность использования групповых адресов. Существуют 5 классов IP-адресов, отличающихся количеством бит в сетевом номере и номере конечной установки (узла). Класс адреса определяется значением его первого октета.

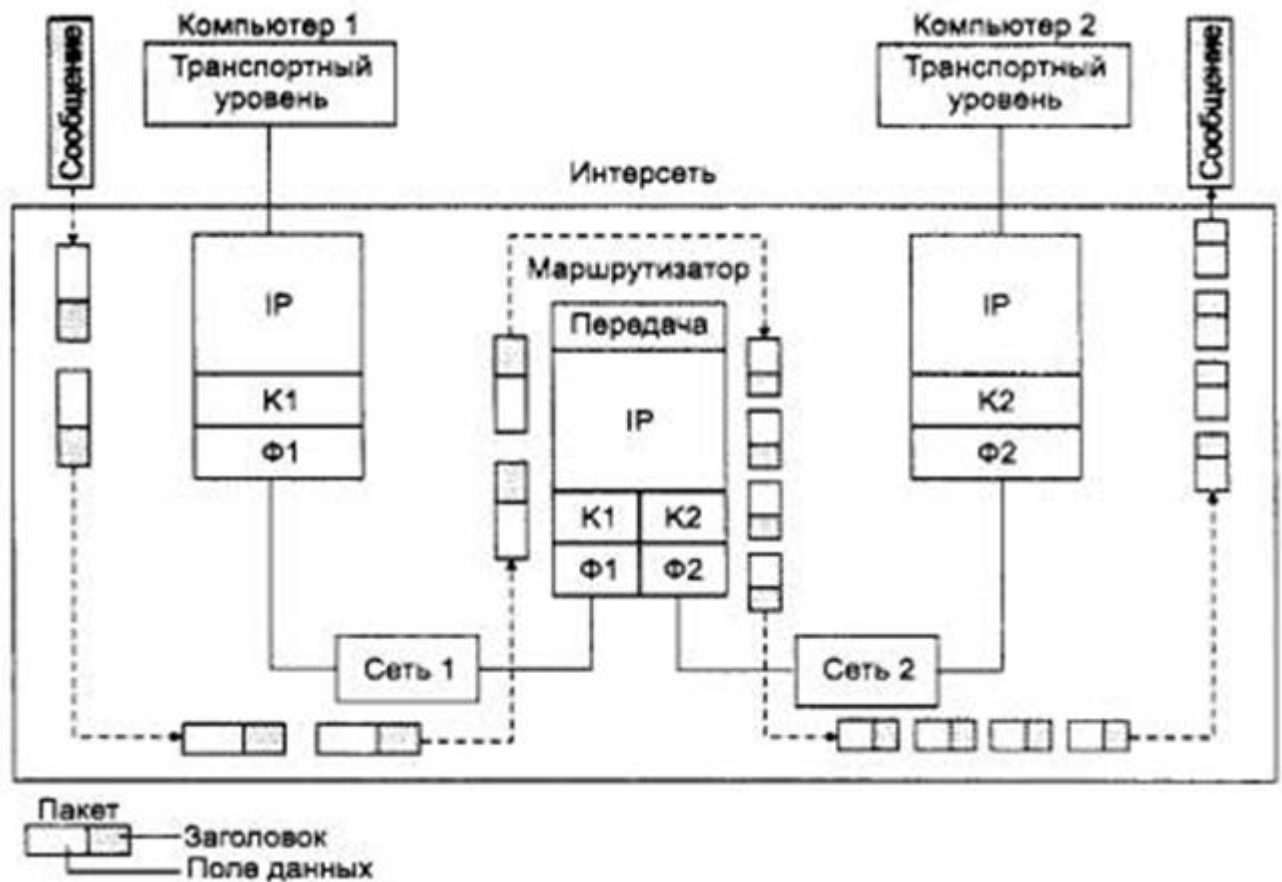
Поле **Вспомогательные параметры IP (опции IP) (Options)** — определяет наличие дополнительных услуг, имеет переменную длину и может присутствовать или отсутствовать в межсетевой дейтаграмме.

Поле **Заполнитель (Padding)** применяется для выравнивания заголовка на 32-разрядную границу.

#### Фрагментация

IP протокол реализует операции сборки и разборки пакетов, связанные с использованием сетей, в которых применяются форматы меньших длин, чем в пакетах получаемых от транспортного уровня. Формат IP-пакета согласуется с форматами пакетов используемых сетей.

Части, на которые разделяется дейтаграмма (IP-пакет), называются фрагментами, а сам процесс разделения - *фрагментацией*.



<="" p="">

Рис. 3.4. [Фрагментация IP-пакетов при передаче между сетями с разными MTU](#)

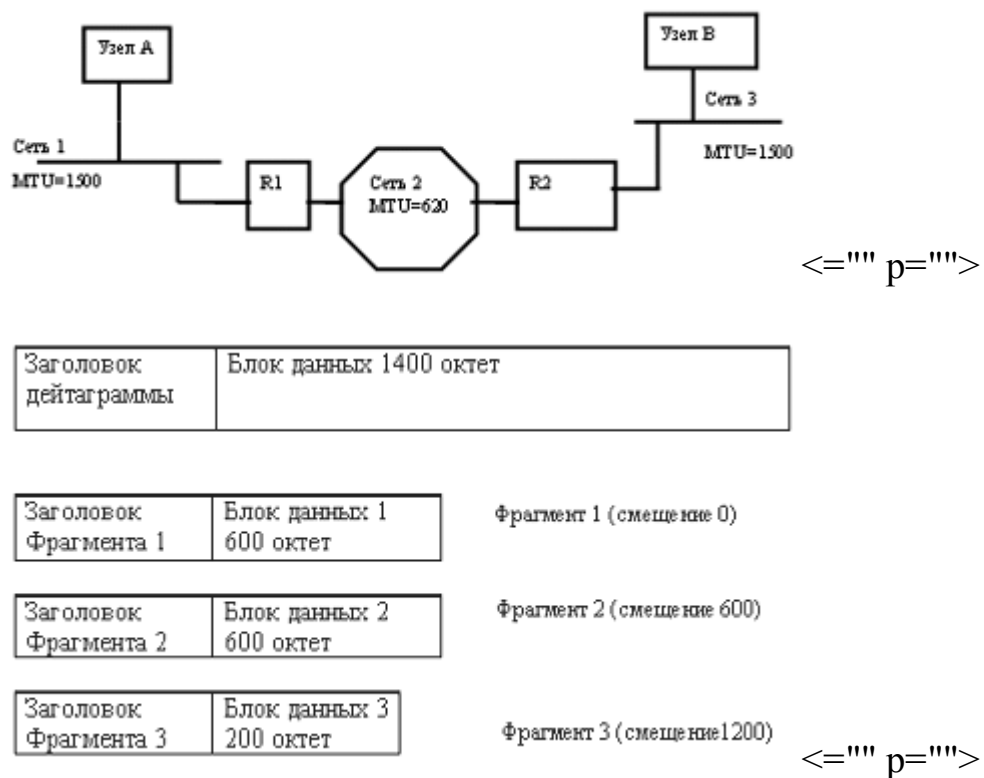
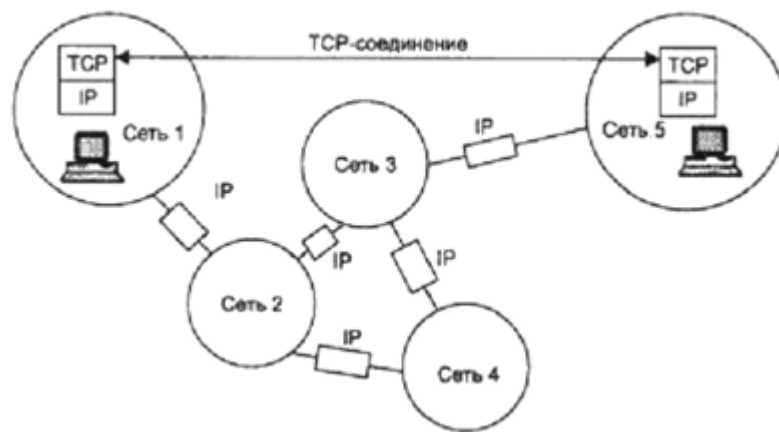


Рис. 3.5. Пример выполнения фрагментации в сети

Шлюз подключается к двум или более сетям, каждая из которых воспринимает этот шлюз как хост-ЭВМ. Поэтому шлюз имеет физический интерфейс и специальный IP-адрес в каждой из подключаемых сетей. Передача пакетов требует от шлюза определение IP-адреса следующего шлюза или, на последнем участке, IP-адреса хост-машины, которой направляется IP-пакет. Функция шлюза, которая обычно называется маршрутизацией, основана на анализе специальных маршрутных таблиц (матриц маршрутов), которые находятся в специальной базе данных. База данных в каждом из шлюзов должна постоянно обновляться, чтобы отражать текущую топологию сети Internet.



<="" p="">

Рис. 3.6. Пример TCP-соединения

В протоколе IP используются как статический, так и динамический способы маршрутизации. **Статическая маршрутизация** используется в оконечных установках локальных сетей, а также в сетях с ограниченным числом абонентов. Маршрутная таблица имеет ограниченный объем и содержит лишь данные о соседних оконечных установках. Пакеты к другим адресатам направляются в маршрутизатор, имеющий существенно больше данных об адресатах сети. Таким образом, маршрутизация в этом случае осуществляется на основе неполной информации.

При использовании **динамической маршрутизации** осуществляется постоянная корректировка маршрутных таблиц на основе данных, содержащихся в служебных сообщениях, которыми маршрутизаторы обмениваются между собой. При определении оптимальных маршрутов используются два класса протоколов. Первый класс протоколов основан на подсчете числа промежуточных ретрансляций в маршрутизаторах и не учитывает реальную пропускную способность каналов передачи данных между маршрутизаторами. В протоколах второго типа оптимизация маршрутов осуществляется на основе измерения времени задержки пакетов, что обеспечивает лучшие условия для выравнивания нагрузки в сети, но приводит к усложнению реализации.

В процессе передачи данных в сети Internet может возникнуть необходимость передать некоторые управляющие сообщения отправителю, например, о недостижимости адресата, истечении времени жизни, о возникновении ошибки в заголовке, а также о переадресации межсетевой дейтаграммы. Для этого используется протокол **ICMP**,

который является составной частью протокола IP и должен быть реализован в каждом межсетевом модуле IP.

### **Контрольные вопросы:**

1. Пример фрагмента сети Интернет.
2. Основные протоколы семейства TCP/IP.
3. Пример передачи сообщений в сети Internet на основе механизма инкапсуляции.
4. Назначение протоколов канального уровня TCP/IP.
5. Основные функции и характеристики протокола IP.
6. Основные механизмы протокола IP.
7. Структура IP-пакета. Функциональное назначение полей заголовка.
8. Примеры выполнения фрагментации IP-пакетов в сети Internet.
9. Назначение вспомогательного протокола ICMP.

## **4. Принципы и алгоритмы маршрутизации в Интернет**

### **4.1. Проблема маршрутизации в сети Internet**

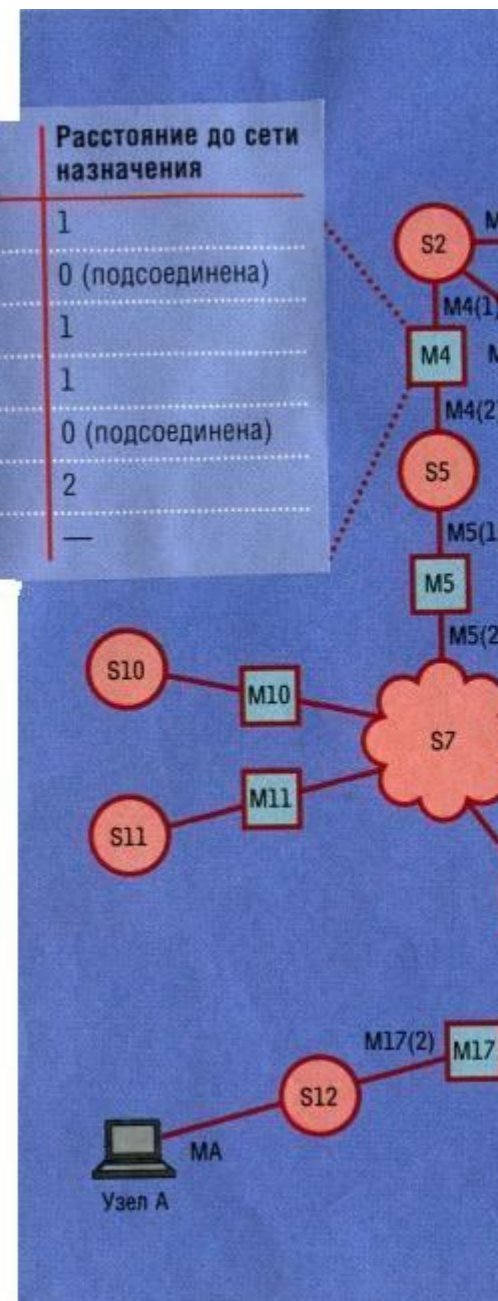
В архитектуре TCP/IP сети соединяются друг с другом коммутаторами IP-пакетов, которые называются шлюзами или IP-маршрутизаторами.

Основная задача IP-маршрутизатора — определение по специальному алгоритму адреса следующего IP-маршрутизатора.

Для решения этой задачи каждый IP-маршрутизатор должен располагать матрицей маршрутов (специальной базой данных, обеспечивающей маршрутизацию), которую необходимо регулярно обновлять.

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|-----------------------|-----------------------------------------|-------------------------------|-------------------------------|
| S1                    | M1(2)                                   | M4(1)                         | 1                             |
| S2                    | —                                       | M4(1)                         | 0 (подсоединена)              |
| S3                    | M1(2)                                   | M4(1)                         | 1                             |
| S4                    | M2(1)                                   | M4(1)                         | 1                             |
| S5                    | —                                       | M4(2)                         | 0 (подсоединена)              |
| S6                    | M2(1)                                   | M4(1)                         | 2                             |
| Default               | M5(1)                                   | M4(2)                         | —                             |

Пример таблиц маршрутизации в составной сети



Алгоритм маршрутизации является тем фундаментом, на котором строится вся работа базовой сети с архитектурой ТСР/ІР. Обеспечение надёжных сетевых услуг требует определённой динамики маршрутизации. Неожиданные изменения в связности базовой сети должны рассматриваться как обычные явления и соответствующим образом обрабатываться, так же как и перегрузки отдельных направлений и каналов.

Существует ряд требований, которые следует учитывать при выборе приемлемого алгоритма маршрутизации:

- алгоритм маршрутизации должен распознавать отказ и восстановление каналов связи или других ІР-маршрутизаторов и переключаться на другие, подходящие маршруты. Время переключения маршрутов должно быть

меньшим, чем типичный тайм-аут пользователя протокола ТСР (примерно 1 мин);

- алгоритм должен исключать образование циклов, петель и эффекта «пинг-понг» в назначаемых маршрутах как между соседними IP-маршрутизаторами, так и для удалённых IP-маршрутизаторов. Существование вышеперечисленных эффектов не должно превышать типичного тайм-аута пользователя протокола ТСР (примерно 1 минута);
- нагрузка, создаваемая управляющими сообщениями, которые необходимы для работы алгоритма маршрутизации, не должна ощутимо ухудшать или нарушать нормальную работу сети. Изменение состояния сети, которое может прервать нормальную работу в некоторой локальной области сети, не должно оказывать воздействия на удалённые участки;
- поскольку размеры сети постоянно увеличиваются, необходимо обеспечить эффективное использование сетевых ресурсов, например, изменение матриц маршрутов выполнять по частям, передавая по глобальным сетям только дополнения к базам данных по маршрутизации;
- размер базы данных по маршрутизации не должен превышать некоторой константы, не зависящей от топологии сети, умноженной на количество узлов и на среднюю связность сети. Хорошая реализация не должна требовать хранения полной базы данных по маршрутизации в каждом IP-маршрутизаторе;
- если используются метрики, основанные на достижимости узла и задержке в доставке пакета, то они не должны зависеть от прямой связности со всеми другими IP-маршрутизаторами или от использования механизмов широковещательной передачи, специфичных для некоторых сетей. Процедуры опроса не должны вносить существенных дополнительных расходов;
- маршруты по умолчанию следует использовать в качестве первоначальных предположений о маршрутизации, чтобы затем выбирать окончательное направление передачи.

Кроме перечисленных выше задач IP-маршрутизатор должен обеспечивать эффективное распределение собственных ресурсов как по пропускной способности каналов, так и по объёму буферных ЗУ, используемых для хранения пакетов, ожидающих передачу. Самая очевидная стратегия «первым пришёл — первым обслужен» (FCFS — First Come First Served) может оказаться неприемлемой в условиях перегрузки сети.

Так, например, нельзя допустить, чтобы высокоскоростной канал захватил весь объём буферных ЗУ, ничего не оставив низкоскоростному каналу. В хороших алгоритмах обязательно должно учитываться поле «Тип сервиса» заголовка IP-пакета; IP-маршрутизатор может назначить

больший приоритет IP-пакетам, передающим управляющую или служебную информацию.

Наконец, алгоритм маршрутизации должен обеспечивать надёжный алгоритм определения состояния каждого канала связи и узла в базовой сети и, если требуется, состояние хост-ЭВМ.

Для этого нужен, по крайней мере, протокол канального уровня, предполагающий периодический обмен кадрами через каждый канал связи.

Однако этого часто оказывается недостаточно, поэтому дополнительно требуется специальный механизм в алгоритмах маршрутизации.

По техническим, административным, географическим, а также иногда и политическим соображениям IP-маршрутизаторы группируются в так называемые «автономные системы».

IP-маршрутизаторы, входящие в одну автономную систему, контролируются одной организацией, обеспечивающей их сопровождение, и используют общие для данной автономной системы алгоритмы маршрутизации.

**Определение.** Конкретный вариант протокола маршрутизации, действующий внутри одной автономной системы, называется внутренним протоколом маршрутизации (IGP — Interior Gateway Protocol).

Возможно, что некоторому IP-пакету, чтобы достичь места назначения, придётся пройти через IP - маршрутизаторы двух или более автономных систем. Поэтому автономные системы должны иметь возможность обмениваться информацией о своём состоянии.

**Определение.** Протокол для обмена служебной информацией между автономными системами называется внешним протоколом маршрутизации (EGP — Exterior Gateway Protocol).

Каждый IP-маршрутизатор должен обеспечить реализацию протоколов физического, канального, межсетевого уровней, а также протоколы доступа к сети. В качестве последних используются протоколы Ethernet, Frame Relay, ATM, SLIP, PPP и ряд других, а для сетей с архитектурой ISO протокол X.25/2 (LAP-B).

Кроме того, IP-маршрутизатору необходима реализация некоторого алгоритма выбора маршрута по таблице маршрутизации, а также алгоритма её обновления.

Процедура выбора пути заложена в протоколе IP, причём IP-уровень не знает всего пути, а владеет лишь информацией о том, какому IP-

маршрутизатору передать IP-пакет с конкретным адресом места назначения.

Просмотр таблицы маршрутизации происходит в три этапа:

- ищется соответствие адреса, записанного в IP-пакете, адресу места назначения в маршрутной таблице. В случае успеха пакет посылается соответствующему IP-маршрутизатору или непосредственно хост-ЭВМ. Связи “точка-точка” выявляются именно на этом этапе;
- ищется соответствие адреса, записанного в IP-пакете, адресу некоторой региональной сети места назначения (одна запись в таблице маршрутизации соответствует всем хостам, входящим в данную региональную сеть). В случае успеха система действует так же, как и в предыдущем пункте;
- ищется маршрут «по умолчанию», если таковой предусмотрен; дейтаграмма посылается в соответствующий маршрутизатор.

Существуют **статические** и **динамические** алгоритмы обновления таблицы.

**Статический** алгоритм есть способ маршрутизации, не изменяющийся при изменении топологии и состояния сети. Примерами являются алгоритмы случайной и лавинной маршрутизации.

**Случайная маршрутизация** — передача данных из узла в любом, случайным образом выбранном направлении, кроме направления, по которому данные поступили в узел. Данные, совершая «блуждания» по сети с конечной вероятностью когда-либо достигают адресата.

**Лавинная маршрутизация** — передача данных из узла во всех направлениях, кроме того, по которому поступили данные. Очевидно, что хотя бы одно направление обеспечит доставку пакета за минимальное время, т.е. лавинная маршрутизация гарантирует малое время доставки.

Шлюзы, входящие в состав одной автономной системы, могут выполнять алгоритм **динамической** маршрутизации — протоколы на основе алгоритма Беллмана-Форда и протоколы на основе алгоритма Дейкстры.

Каждой дуге графа ставится в соответствие действительное число, называемое длиной дуги; тогда длина пути определяется суммой длин составляющих его дуг.

Обычно это число переприёмов или средняя задержка пакетов, но возможны и другие метрики, например, пропускная способность канала связи, надёжность.

Шлюзы, работающие по **алгоритму Беллмана-Форда**, хранят вектор длин кратчайших маршрутов до всех сетей, входящих в состав

объединённой сети.

Периодически каждый шлюз передаёт свой вектор соседним шлюзам автономной системы, а элементы вектора, принятого от соседнего шлюза, складываются с длинами исходящих линий связи.

На основе полученной таблицы строится новый вектор длин кратчайших маршрутов — алгоритм Беллмана-Форда (DV — алгоритм Distance Vector).

Протоколы на основе DV-алгоритма достаточно просто реализуются, требуют мало памяти и процессорного времени, однако они обладают рядом общих недостатков. При увеличении количества сетей, входящих в состав автономной системы, резко возрастает количество передаваемой информации, т.к. DV-алгоритм требует, чтобы все шлюзы периодически передавали свои векторы длин маршрутов.

Шлюзы, работающие по **алгоритму Дейкстры** (Shortest Path First — SPF-алгоритм), сначала определяют кратчайшие маршруты по всем сетям автономной системы. Для этого в каждом шлюзе строится полное дерево кратчайших путей с корнем в данном шлюзе.

Процедура построения дерева кратчайших путей использует принцип, согласно которому в дерево кратчайших путей первой включается дуга с наименьшей длиной, поэтому алгоритм Дейкстры часто называют первым кратчайшим путем.

После того как в шлюзе построено дерево кратчайших путей, изменения характеристик линий связи, определяющих длины соответствующих дуг графа, изменения топологии сети приводят к небольшим дополнительным вычислениям для корректирования дерева кратчайших путей.

Шлюзы обмениваются только сведениями о длинах исходящих линий связи, а не векторами длин маршрутов, как в случае алгоритма Беллмана-Форда. Размер корректирующих пакетов со служебной информацией для маршрутизации мал и не зависит от числа сетей в автономной системе. Каждый шлюз посылает такие пакеты с помощью лавинной маршрутизации. При появлении в сети нового шлюза или включении новой линии связи изменения в топологии сети не учитываются при маршрутизации в течение некоторого времени для того, чтобы информация о происшедших изменениях успела достигнуть всех шлюзов автономной системы.

В целом, алгоритм Дейкстры, по сравнению с алгоритмом Беллмана-Форда, обеспечивает более реальную оценку ситуации в сети, более быструю реакцию на важные изменения в сети (такие, как включение новой линии связи) и уменьшает заикливание пакетов; однако алгоритм Дейкстры сложнее в реализации и требует в несколько раз больше памяти.

## 4.2. Внутренние протоколы маршрутизации

Поле **Протокол GGP** (Gateway to Gateway Protocol, RFC 823) был разработан и реализован фирмой BBN для первых экспериментальных шлюзов сети Internet.

Он до сих пор используется в шлюзах фирмы BBN LSI/11, хотя считается, что GGP имеет серьёзные недостатки и позднее был заменён на алгоритм SPF.

Алгоритм протокола GGP определяет маршрут с минимальным числом переприёмов, т.е. его мерой длины является просто число транзитных участков сети между парами шлюзов.

Он реализует распределённый алгоритм кратчайшего пути, который требует глобальной сходимости маршрутных таблиц после изменений в топологии или связности.

**Протокол RIP** (Routing Information Protocol, RFC 1058, 1581, 1582, 1724) часто используется для класса протоколов маршрутизации, базирующихся на протоколах XNS (Xerox Network System — сетевая система Хероx) фирмы Хероx.

Реализация протокола RIP для семейства протоколов TCP/IP широко доступна, поскольку входит в состав программного обеспечения ОС UNIX, например, FreeBSD или Linux. В силу своей простоты протокол RIP имеет наибольшие шансы превратиться в **«открытый»** протокол IGP, т.е. протокол, который может использоваться для совместной работы шлюзов, поставляемых разными фирмами.

В качестве метрики маршрутизации RIP использует число скачков (шагов) до цели. Такой вид метрики не учитывает различий в пропускной способности или загруженности отдельных сегментов сети.

Каждому маршруту ставится в соответствие таймер **тайм-аута** и **«сборщик мусора»**. Таймер тайм-аута сбрасывается каждый раз, когда маршрут инициализируется или корректируется. Если со времени последней коррекции прошло 3 минуты или получено сообщение в том, что вектор расстояния равен 16, маршрут считается закрытым, но запись о нём не стирается, пока не истечёт время «уборки мусора» (2 минуты). При появлении эквивалентного маршрута переключение на него не происходит.

Протокол RIP достаточно простой, но не лишённый недостатков:

- требуется много времени для восстановления связи после сбоя в маршрутизаторе (минуты); в процессе установления режима возможны циклы;
- число шагов — важный, но не единственный параметр маршрута, да и 15 шагов — не предел для современных сетей.

**Протокол “HELLO”.** Программное обеспечение Fuzzball для шлюза LSI/11 включает в себя реализацию протокола IGP под названием “HELLO”. В отличие от RIP в нём критерием выбора маршрута служит время, а не расстояние, поэтому “HELLO” требует достаточно точной синхронизации служб времени шлюзов.

**Протокол OSPF (Open Shortest Path First, RFC 1850, 1583, 1584, 1587)** представляет собой протокол состояния маршрута, причём в качестве метрики используется коэффициент качества обслуживания.

Каждый маршрутизатор обладает полной информацией о состоянии всех интерфейсов шлюзов автономной системы. Определяющими являются три характеристики: задержка, пропускная способность и надёжность.

Преимущества OSPF:

- для каждого адреса может быть несколько маршрутных таблиц, по одной на каждый вид IP-операции;
- каждому интерфейсу присваивается безразмерная цена, учитывающая пропускную способность, время транспортировки сообщения; каждой IP-операции может быть присвоена своя цена;
- при существовании эквивалентных маршрутов OSPF распределяет поток равномерно по этим маршрутам;
- при связи «точка–точка» не требуется IP-адрес для каждого из концов;
- применяется мультикастинг вместо широковещательной адресации, что снижает загрузку не вовлечённых в обмен сегментов.

Недостатки OSPF — трудно получить информацию о предпочтительности каналов для узлов, поддерживающих другие протоколы или имеющих статическую маршрутизацию.

**Протокол IS-IS.** Учитывая, с одной стороны, широкое распространение сетей с архитектурой TCP/IP, с другой стороны, повышенное внимание правительственных и коммерческих организаций к архитектуре ЭМВОС; ожидается, что архитектуры TCP/IP и ЭМВОС будут существовать долгое время вместе. Поэтому возникает необходимость в шлюзах, способных маршрутизировать одновременно IP- и ЭМВОС-трафик.

Протокол IS-IS (Intermediate System to Intermediate System Protocol, RFC 1195) обеспечивает поддержку понятий IP-подсети, переменной маски подсети, маршрутизацию на основе значения поля «Тип сервиса» в заголовке IP-пакета и понятие внешнего маршрута.

Протокол IS-IS является динамическим протоколом маршрутизации, построенным на основе SPF-алгоритма.

## **4.3. Внешние протоколы маршрутизации**

Внешние протоколы маршрутизации предназначены главным образом для связи между автономными (независимыми) системами.

**Протокол EGP** (Exterior Gateway Protocol) один из известных протоколов этого типа (RFC 827, 888, 904, 911, 1092, 1093).

Документ RFC 827, предложивший первую модель шлюза для взаимодействия со шлюзами других автономных систем, и документ RFC 888, представляющий собой развитие этой модели, накладывали существенное ограничение на топологию сети Internet, предполагая древовидную двухуровневую структуру, корнем которой является так называемая «магистральная» автономная система, состоящая из «магистральных» шлюзов.

Главным преимуществом такой модели считалась невозможность образования в древовидной топологической структуре циклических маршрутов между автономными системами.

С помощью протокола EGP шлюзы могут снабжать друг друга информацией о достижимости соседних шлюзов и о маршрутах к соседним шлюзам. При этом динамическое вычисление маршрутов выполняется только шлюзами магистральной автономной системы, и затем результаты могут быть сообщены немагистральным шлюзам.

Немагистральные шлюзы также могут предоставлять маршрутную информацию магистральным и немагистральным шлюзам, но они не имеют права передавать дальше маршруты, вычисленные на основе информации, полученной от других шлюзов. Это ограничение часто называется ограничением на распространение информации «третьей группы».

Протокол EGP включает в себя механизм определения достижимости соседей (соседними называются шлюзы, совместно выполняющие протокол EGP), контроля достижимости и обмена информацией в форме обновляющих сообщений. Цель использования алгоритма достижимости — убедиться в том, что сосед работает и может поставлять надёжную информацию.

Не менее важной является задача фильтрации информации перед тем, как отправлять её другим шлюзам, чтобы избежать лишних изменений базы данных.

Как правило, локальные шлюзы передают по внешнему протоколу только сведения, касающиеся своих автономных систем, чтобы не увеличивать без необходимости трафик в сетях.

**Протокол BGP** (Border Gateway Protocol, RFC 1267) — это протокол маршрутизации между автономными системами в сети Internet; он построен на основе опыта, накопленного при эксплуатации протокола

EGP.

Главная цель BGP — сократить транзитный трафик. Протокол BGP использует расширенное понятие автономной системы. В данном случае внутри автономной системы шлюзы могут использовать несколько различных протоколов маршрутизации и несколько метрик. Однако внутри автономной системы должен существовать единый план маршрутизации, позволяющий рассматривать автономную систему как единое целое.

В зависимости от того, с каким трафиком имеет дело автономная система, она причисляется к одной из следующих категорий:

- тупиковая автономная система, имеющая единственное соединение с другими автономными системами; фактически, тупиковая система имеет дело только с локальным трафиком;
- многовходовая автономная система. Эта система имеет более одного соединения с другими автономными системами, но она отказывается поддерживать транзитный трафик;
- транзитная автономная система, которая имеет более одного соединения с другими автономными системами и предназначена для поддержания обоих видов трафика.

Протокол BGP использует в качестве транспортного протокола протокол TCP.

Хост-ЭВМ, выполняющие протокол BGP, не обязательно должны одновременно являться шлюзами.

Хост-ЭВМ, не являющаяся шлюзом, может обмениваться маршрутной информацией со шлюзами при помощи протокола EGP или внутреннего протокола маршрутизации.

Эта хост-ЭВМ может затем использовать протокол BGP для обмена маршрутной информацией с граничным шлюзом другой автономной системы.

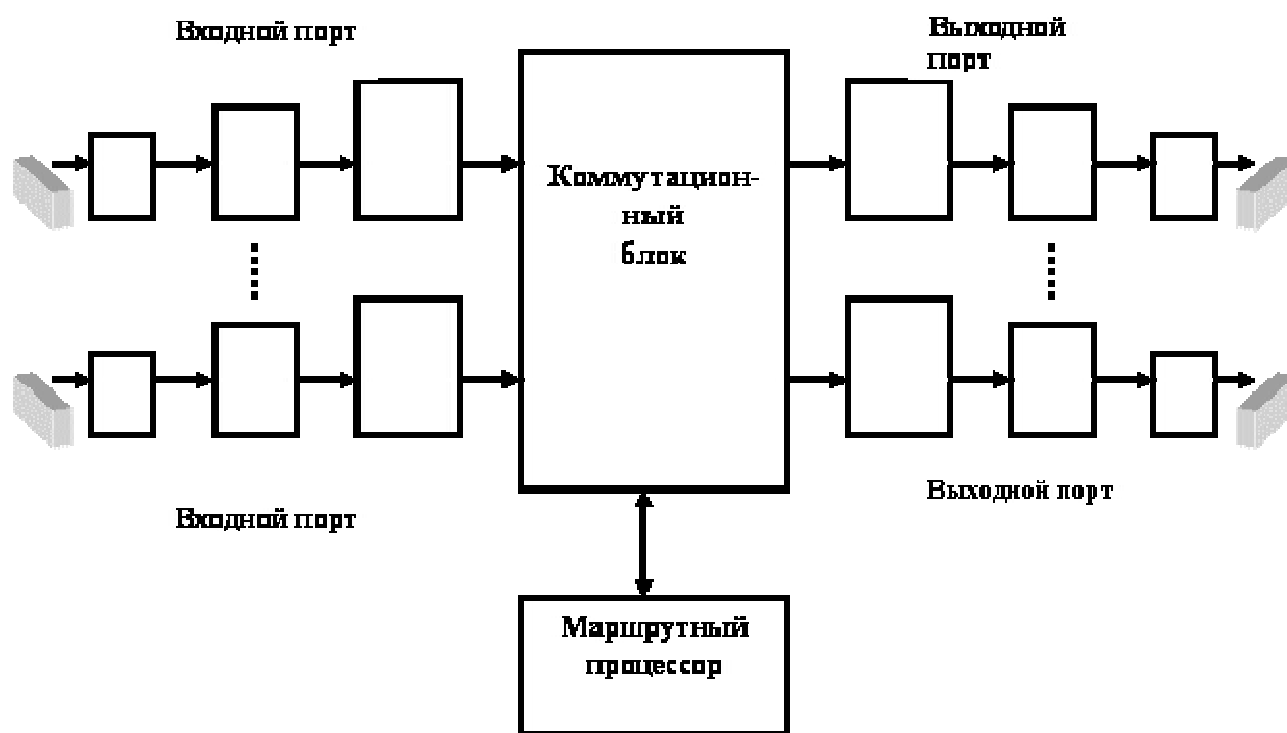


Рис. 4.1. Архитектура маршрутизатора

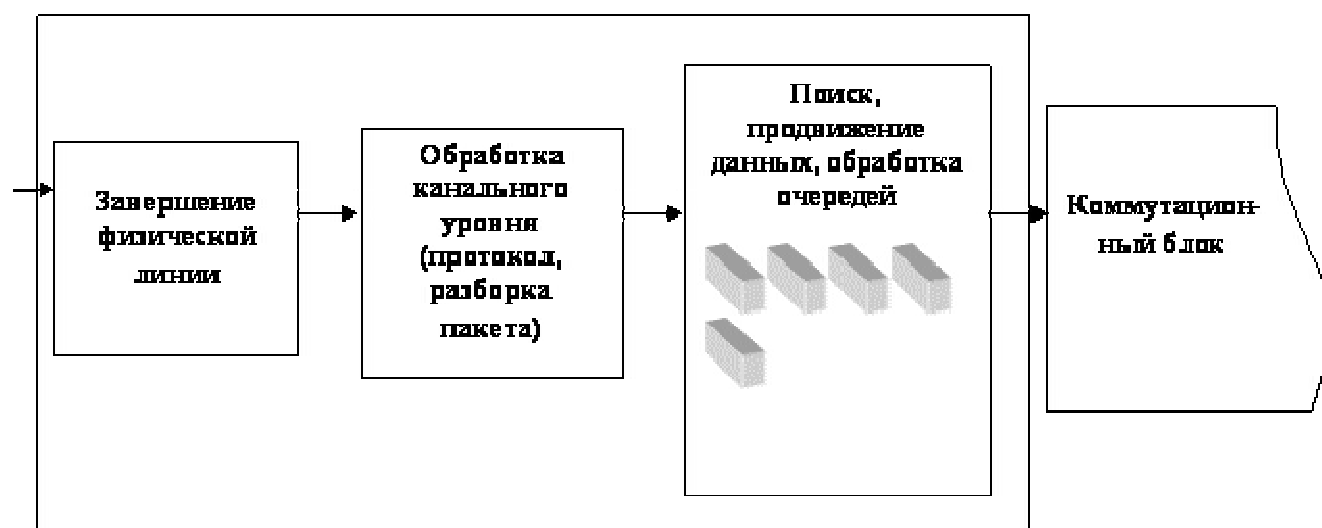


Рис. 4.2. Обработка на входном порту

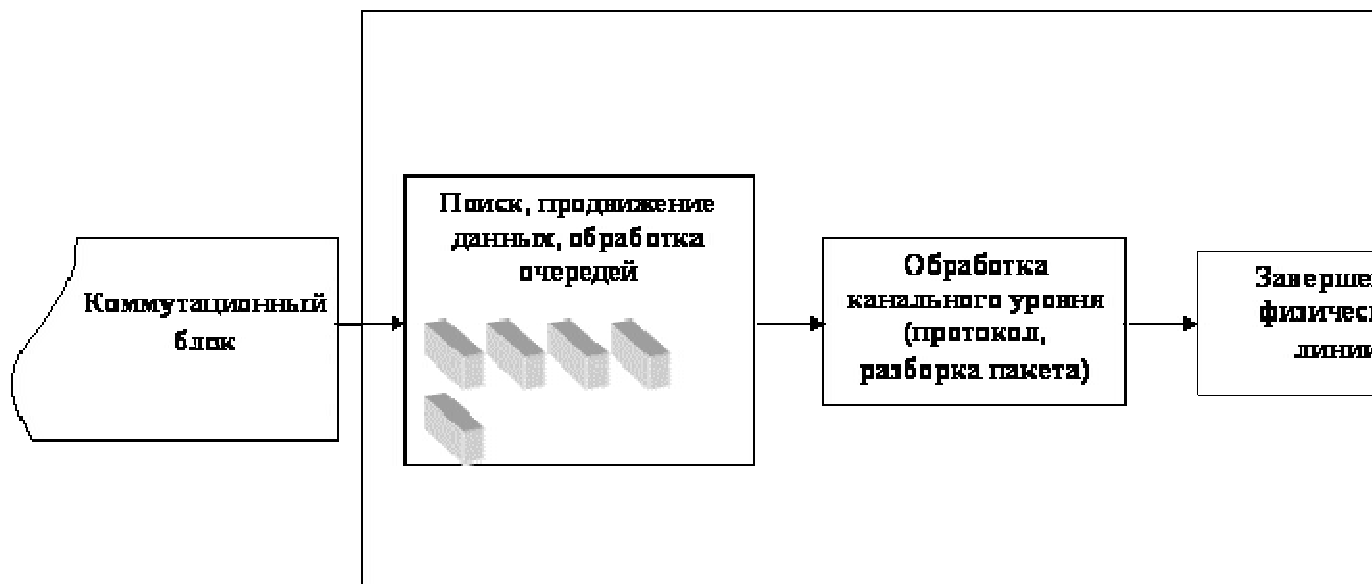
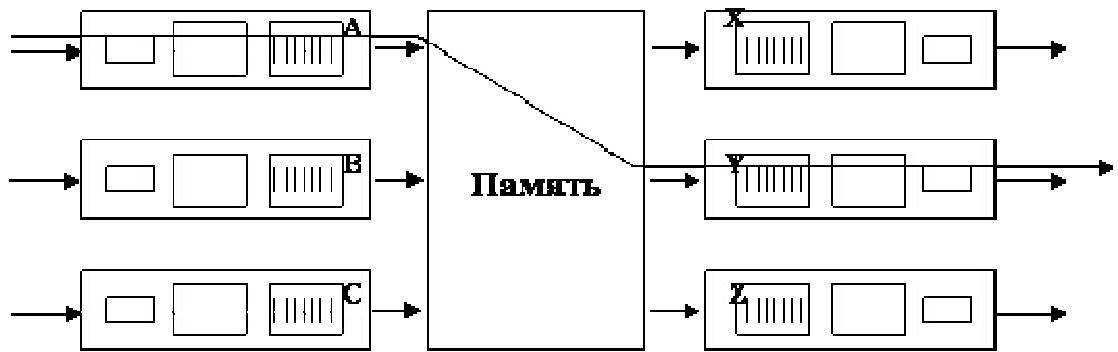
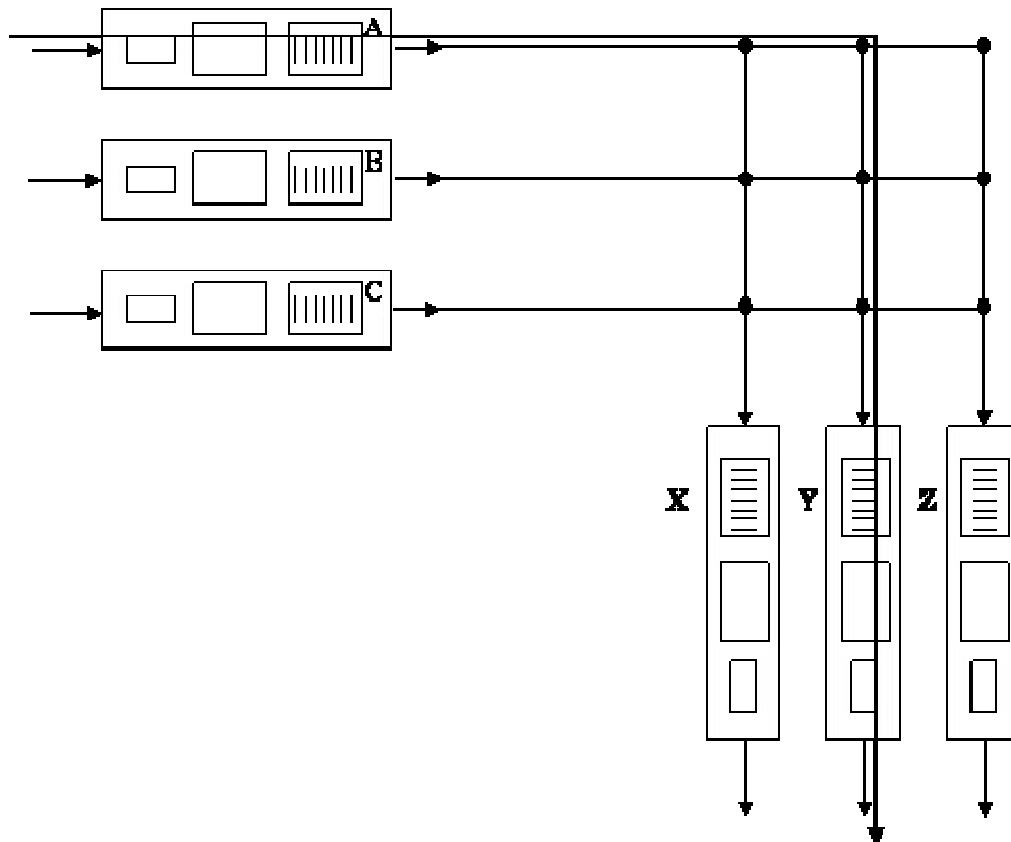


Рис. 4.3. Обработка данных на выходном порту

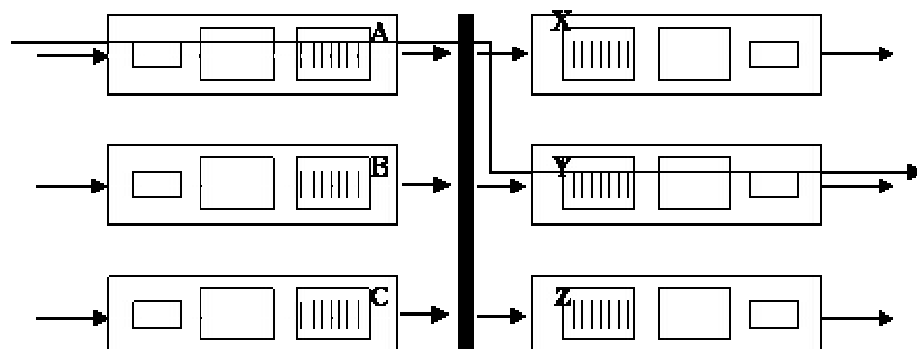
### Память



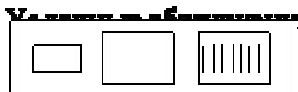
### Матричный коммутатор



### Шина



### Устройства ввода/вывода

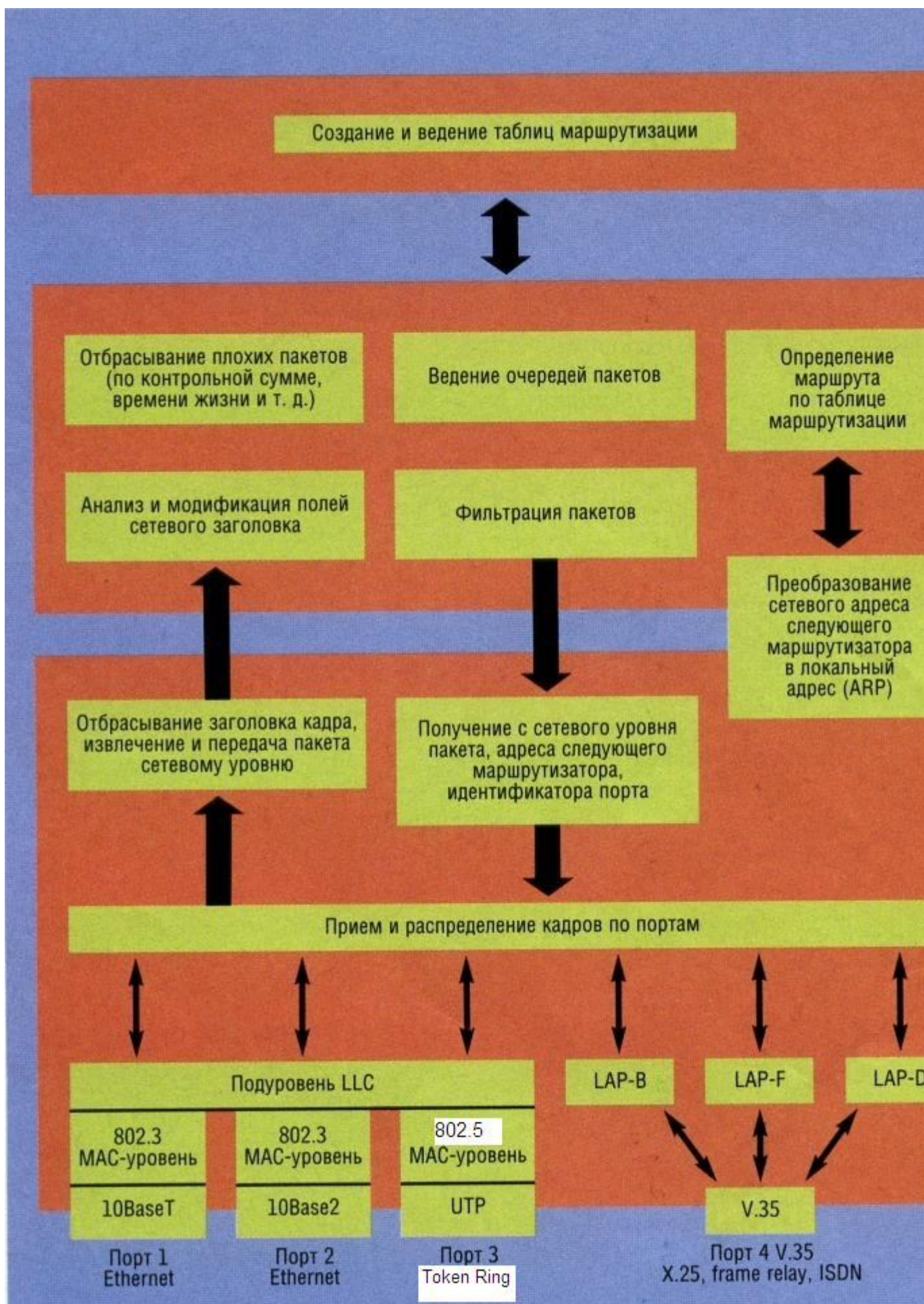


Выходной порт



Рис. 4.4. Три метода коммутации

**Функциональная модель маршрутизатора**



**Контрольные вопросы:**

1. Поясните основные свойства алгоритма маршрутизации.
2. Дайте пояснение понятий «автономная система», «внутренние и внешние протоколы маршрутизации».
3. Назначение маршрутной таблицы.
4. Опишите статический алгоритм обновления таблицы.
5. Дайте объяснение динамическому алгоритму обновления таблицы маршрутизации.
6. Объясните принцип алгоритма Беллмана-Форда.
7. Объясните принцип алгоритма Дейкстры.
8. Назначение протоколов RIP и GGP.
9. Назначение протоколов HELLO, OSPF и IS-IS.
10. Основные положения внешнего протокола EGP.
11. Назначение внешнего протокола BGP.

## **5. Организация передачи трафика реального времени по сети Интернет. Протокол RTP. Мультимедиа**

*В типичной среде реального времени отправитель генерирует пакеты с постоянной скоростью.*

*Они отправляются через одинаковые интервалы времени, проходят через сеть и принимаются получателем, воспроизводящим данные в реальном времени при их получении.*

*Однако ввиду изменения времени задержки при передаче пакетов по сети, они могут прибывать через нерегулярные интервалы времени.*

*Для компенсации этого эффекта поступающие пакеты буферизуются, придерживаются на некоторое время и затем предоставляются с постоянной скоростью программному*

*обеспечению, генерирующему вывод.*

*Поэтому для функционирования протокола реального времени необходимо, чтобы каждый пакет содержал временную метку — тогда получатель сможет воспроизвести поступающие данные с той же скоростью, что и отправитель.*

## **5.1. Протокол передачи видео- и аудиоинформации в реальном масштабе времени — RTP**

### ***Почему нужен RTP***

В Интернет возможна потеря пакетов, изменение их порядка в процессе транспортировки, а также вариация времени доставки в достаточно широких пределах.

В приложениях реального времени (аудио- и видеоконференции, живое видео, удаленная диагностика в медицине, компьютерная телефония, игры, мониторинг в реальном времени и др.) отправитель генерирует поток данных с постоянной скоростью, а получатель (или получатели) должен предоставить эти данные приложению с той же самой скоростью.

Протокол транспортного уровня — TCP не подходит для приложений реального времени.

1. TCP позволяет установить соединение только между двумя конечными точками, следовательно, он не подходит для многоадресной передачи.

2. TCP предусматривает повторную передачу потерянных сегментов, прибывающих, когда приложение реального времени уже их не ждет.

3. TCP не имеет удобного механизма привязки информации о синхронизации к сегментам.

Для согласования требований приложений реального времени с возможностями Интернет был разработан транспортный протокол реального времени — **RTP** (Real-Time Transport Protocol).

Протокол RTP (**RFC 1889** — A Transport Protocol for Real-Time Applications” Н. Schulzrinne, S. Casner, R. Frederick, V. Jacobson) обеспечивает в IP-сетях доставку адресатам аудио- и видеопотоков в масштабе реального времени.

Согласно стеку рекомендаций **H.323**, в сетях с негарантированной полосой пропускания с целью минимизации задержек и максимального использования имеющейся полосы пропускания для передачи аудио- и видеопотоков, а также сигнализации **RAS** применяется протокол **User**

## **Datagram Protocol (UDP).**

Этот протокол задействует механизм многоадресной рассылки (IP Multicast) для негарантированной доставки звука и видео определенному числу пользователей.

Поверх IP Multicast работает RTP, который создает необходимые условия для нормального воспроизведения полученных потоков на абонентских терминалах.

### ***Принципы построения протокола RTP***

RTP идентифицирует тип и номер пакета, устанавливает в него метку синхронизации.

Пакеты RTP содержат также идентификатор отправителя, указывающий, кто из участников генерирует данные, и информацию о характере содержимого пакета, например, о типе кодировки видеоданных.

На основе этой информации приемный терминал синхронизирует звук, видео и данные, осуществляет их последовательное и непрерывное воспроизведение.

Корректное функционирование RTP возможно при наличии в абонентских терминалах механизмов буферизации принимаемой информации.

Реализацию функций RTP контролирует транспортный протокол управления передачей в режиме реального времени **RTCP — Real-time Transport Control Protocol** (RFC 1889).

Он также отслеживает качество обслуживания и снабжает соответствующей информацией участников конференции. RTCP использует тот же самый базовый транспортный протокол, что и RTP (обычно UDP), но другой номер порта.

RTP работает поверх UDP и может поддерживать передачу данных в реальном времени между несколькими участниками RTP-сеанса.

RTCP выполняет несколько функций:

1. Обеспечение и контроль качества услуг и обратная связь в случае перегрузки.

Так как RTCP-пакеты являются многоадресными, все участники сеанса могут оценить, насколько хороши работа и прием других участников.

Сообщения отправителя позволяют получателям оценить скорость данных и качество передачи.

Сообщения получателей содержат информацию о проблемах, с которыми они сталкиваются, включая утерю пакетов и избыточную

неравномерность передачи.

Обратная связь с получателями важна также для диагностирования ошибок при распространении. Анализируя сообщения всех участников сеанса, администратор сети может определить, касается данная проблема одного участника или носит общий характер.

Если приложение-отправитель приходит к выводу, что проблема характерна для системы в целом, например, по причине отказа одного из каналов связи, то оно может увеличить степень сжатия данных за счет снижения качества или вообще отказаться от передачи видео — это позволяет передавать данные по соединению низкой емкости.

## 2. Идентификация отправителя.

Пакеты RTCP содержат стандартное текстовое описание отправителя. Они предоставляют больше информации об отправителе пакетов данных, чем случайным образом выбранный идентификатор источника синхронизации.

Кроме того, они помогают пользователю идентифицировать потоки, относящиеся к различным сеансам.

## 3. Оценка размеров сеанса и масштабирование.

Для обеспечения качества услуг и обратной связи с целью управления загруженностью, а также с целью идентификации отправителя, все участники периодически посылают пакеты RTCP.

Частота передачи этих пакетов снижается с ростом числа участников.

При небольшом числе участников один пакет RTCP посылается максимум каждые 5 секунд.

RFC 1889 описывает алгоритм, согласно которому участники ограничивают частоту RTCP-пакетов в зависимости от общего числа участников. Цель состоит в том, чтобы трафик RTCP не превышал 5% от общего трафика сеанса.

При организации видеоконференции каждый участник должен иметь IP-адрес и две пары UDP-портов: для RTP и RTCP, так как звук и изображение передаются как два независимых потока. RTCP-пакеты посылаются независимо для каждой из этих двух сессий.

На уровне RTP не существует какой-либо взаимосвязи между аудио- и видео сессиями. Только RTCP-пакеты несут в себе одни и те же имена участников.

## ***Определения***

**Пакет RTP.** Информационный пакет, содержащий фиксированный заголовок. Один пакет транспортного нижнего уровня, например UDP, обычно содержит один RTP-пакет.

**Поле данных RTP.** Информация, пересылаемая в пакете RTP, например, фрагменты звука или сжатые видео данные.

**Пакет RTCP.** Управляющий пакет, содержащий фиксированный заголовок, сходный с RTP, за которым следуют структурные элементы, зависящие от типа RTCP-пакета. Обычно несколько RTCP-пакетов посылаются как составной RTCP-пакет, вложенный в дейтограмму нижележащего уровня.

**Транспортный адрес.** Комбинация сетевого адреса и порта, которая идентифицирует конечную точку канала (например, IP-адрес и UDP-порт).

Пакеты следуют от транспортного адреса отправителя к транспортному адресу получателя.

**Сессия RTP.** Период с момента установления группой участников RTP-обмена до его окончания.

**Источник синхронизации (SSRC - Synchronization SouRCe).** Источник потока RTP-пакетов, определяется 32-битным числовым SSRC-идентификатором, который записывается в заголовок RTP-пакета и не зависит от сетевого адреса.

Все пакеты от источника синхронизации содержат временную привязку и нумерацию. Эти данные используются принимающей стороной при воспроизведении.

Источниками синхронизации могут служить источники первичного сигнала (микрофоны или видеокамеры), а также RTP-смесители.

SSRC-идентификатор представляет собой случайное число, которое является уникальным для данной RTP-сессии. Если формируется несколько потоков в рамках одной RTP-сессии (например, от нескольких видеокамер), то каждый поток должен быть снабжен уникальным SSRC-идентификатором.

**Информационный источник (CSRC - Contributing SouRCe).** Источник потока RTP-пакетов, который вносит вклад в общий поток, формируемый RTP-смесителем.

Смеситель вставляет список SSRC-идентификаторов, которые идентифицируют отдельные источники, в заголовок RTP-пакетов. Этот список называется CSRC-списком.

Примером приложения может быть аудио-конференция, где смеситель отмечает всех говорящих, чей голос порождает исходящие пакеты. Это

позволяет принимающей стороне идентифицировать говорящего, хотя все пакеты имеют один и тот же SSRC-идентификатор.

**Оконечная система.** Приложение, которое генерирует или воспринимает данные, посылаемые в виде RTP-пакетов. Оконечная система может выступать в качестве одного или нескольких источников синхронизации для конкретной сессии.

Кроме окончательных систем RTP поддерживает трансляторы и смесители, которые рассматриваются как промежуточные системы на уровне RTP.

**Смеситель.** Промежуточная система, которая получает RTP-пакеты от одного или нескольких источников, при необходимости меняет их формат, объединяет и пересылает их адресатам.

Например, смеситель преобразует поток аудио-пакетов в последовательность пакетов, которая соответствует возможностям узкополосного канала.

Так как временная привязка входных пакетов может отличаться, смеситель осуществляет их синхронизацию и генерирует свой собственный поток RTP-пакетов. Таким образом, все посылаемые пакеты имеют в качестве источника синхронизации смеситель.

**Транслятор.** Промежуточная система, которая переадресует RTP-пакеты, не изменяя их идентификаторы источников синхронизации.

Транслятор, создает один исходящий RTP-пакет для каждого поступающего пакета RTP. Этот механизм может изменить формат данных в пакете или использовать иной комплект низкоуровневых протоколов для передачи данных из одного домена в другой.

Например, потенциальный получатель может оказаться не в состоянии обрабатывать высокоскоростной видеосигнал, используемый другими участниками сеанса. Транслятор конвертирует видео в формат более низкого качества, требующий не такой высокой скорости передачи данных.

**Монитор.** Приложение, которое получает RTCP-пакеты, посланные участниками RTP-сессии, в частности диагностические сообщения, производит оценку состояния связи, копит долгосрочную статистику обмена.

### *Заголовок RTP-пакета*

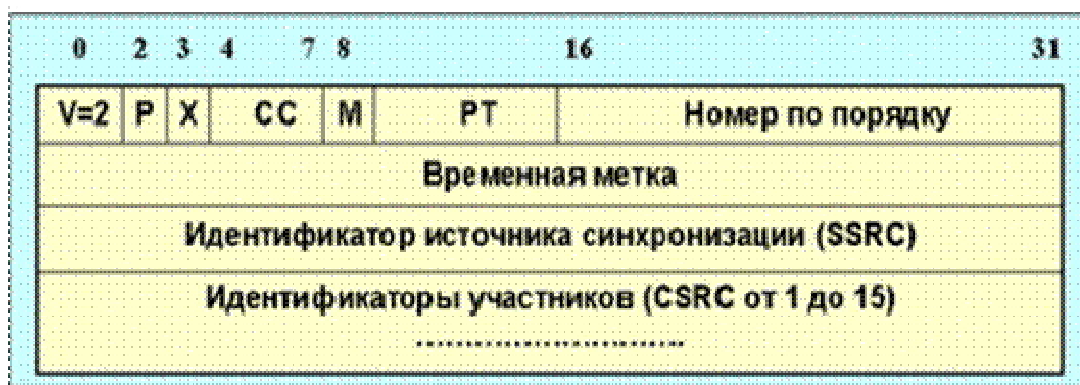


Рис. 5.1. Заголовок пакета RTP

Первые 12 октетов присутствуют во всех RTP-пакетах. Список CSRC-идентификаторов присутствует только тогда, когда пакет формируется смесителем.

**V** (Версия протокола RTP). Текущая версия — вторая.

**P** (Заполнитель). Если P=1, пакет содержит один или более дополнительных октетов-заполнителей в конце поля данных (заполнители не являются частью поля данных).

Последний октет заполнителя содержит число октетов, которые должны игнорироваться.

Заполнитель нужен при использовании некоторых алгоритмов шифрования при фиксированном размере блоков или при укладке нескольких RTP-пакетов в один UDP.

**X** (Расширение). Если X=1, то за основным заголовком следует еще один дополнительный, используемый в экспериментальных расширениях RTP.

**CC** (CSRC count – число CSRC). Это поле содержит количество идентификаторов отправителей, чьи данные находятся в пакете, причем сами идентификаторы следуют за основным заголовком.

**M** (Маркер). Интерпретация маркера определяется профайлом.

Например, предполагается разрешить выделять в потоке пакетов существенные события, такие как границы кадра в случае видео, или в случае голоса выделять начало речи после периода молчания.

**PT** (Тип данных). Идентифицирует тип и формат данных RTP-пакета, включая сжатие и шифрование данных, и определяет интерпретацию поля данных приложением.

**Номер по порядку (Sequence Number).** Каждый источник начинает нумеровать пакеты с произвольного номера, увеличиваемого затем на единицу с каждым посланным пакетом данных RTP.

Это позволяет обнаружить потерю пакетов и определить порядок пакетов с одинаковой отметкой о времени.

Несколько последовательных пакетов могут иметь одну и ту же отметку о времени, если логически они порождены в один и тот же момент, как, например, пакеты, принадлежащие к одному и тому же видеокадру.

**Временная метка (Timestamp).** Поле отметки о времени.

Это поле содержит момент времени, в который первый октет данных полезной нагрузки был создан.

Единицы, в которых время указывается в этом поле, зависят от типа полезной нагрузки. Значение определяется по локальным часам отправителя. Начальное значение временной метки является случайным.

Несколько последовательных RTP-пакетов могут иметь идентичные временные метки, если логически они генерируются одновременно (например, относятся к одному и тому же видеокадру).

**SSRC (Synchronization Source Identifier).** Поле SSRC идентифицирует источник синхронизации.

Этот идентификатор выбирается случайным образом, так чтобы в пределах одной RTP-сессии не было двух равных SSRC-кодов, и не зависит от сетевого адреса. Все приложения должны быть способны выявлять случаи равенства SSRC-кодов.

Если отправитель изменяет свой транспортный адрес, он должен также сменить и SSRC-идентификатор.

В протоколе RTP предусмотрен механизм расширений заголовка, который позволяет модифицировать заголовок и использовать приложения с новыми форматами поля данных.

Протокол RTCP, как и всякий управляющий протокол, значительно сложнее и по структуре, и по выполняемым функциям.

Хотя основу протокола RTCP составляет RTP, он содержит множество дополнительных полей, с помощью которых он реализует свои функции.

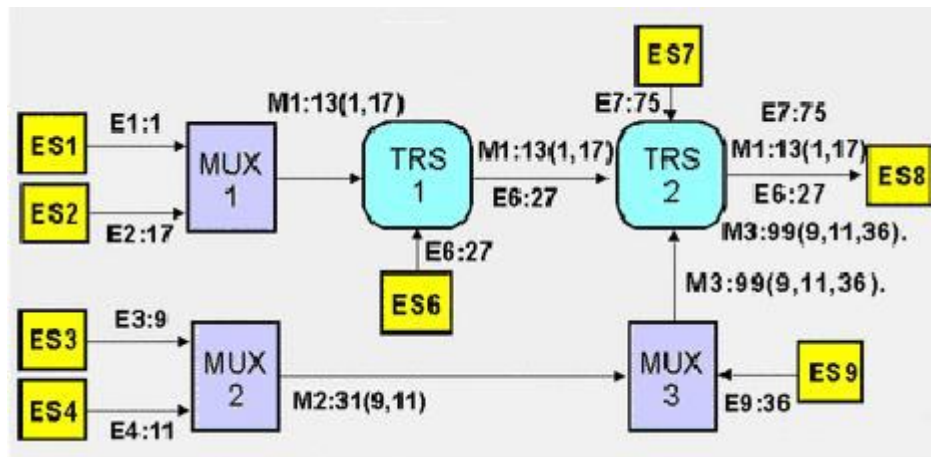


Рис. 5.2. Пример RTP сети с оконечными системами ES, смесителями MUX и трансляторами TRS

**Оконечная система** генерирует или воспринимает данные, посылаемые в виде RTP-пакетов.

Алгоритмы работы отправителя и получателя RTP-пакетов описаны в RFC 1889.

**Смеситель** принимает потоки RTP-данных от одного или нескольких источников и формирует из них один общий поток.

Так как объединяемые потоки не синхронизованы, смеситель производит синхронизацию потоков и формирует свою собственную временную шкалу для исходящего потока. Таким образом, все пакеты данных, переадресованные смесителем, будут помечены SSRC-идентификатором смесителя.

Для того чтобы сохранить информацию об источниках исходных данных, смеситель должен внести SSRC-идентификаторы из заголовков входящих RTP-пакетов в список CSRC-идентификаторов формируемого исходящего пакета, а также должен включить свой собственный SSRC-идентификатор в CSRC-список для данного пакета.

Например, запись "M1:13(1,17)" обозначает пакет, отправленный смесителем MUX1, который идентифицируется случайным значением SSRC 13 и двумя CSRC-идентификаторами 1 и 17, скопированными с SSRC-идентификаторов пакетов оконечных систем ES1 и ES2.

**Транслятор** переадресует RTP-пакеты, не изменяя их SSRC-идентификаторы. Некоторые типы трансляторов передают данные без изменений, другие кодируют данные и соответственно изменяют коды типа данных и временные метки.

Так как объединяемые потоки не синхронизованы, смеситель производит синхронизацию потоков и формирует свою собственную временную шкалу для исходящего потока. Таким образом, все пакеты данных, переадресованные смесителем, будут помечены SSRC-

идентификатором смесителя.

Для того чтобы сохранить информацию об источниках исходных данных, смеситель должен внести SSRC-идентификаторы из заголовков входящих RTP-пакетов в список CSRC-идентификаторов формируемого исходящего пакета, а также должен включить свой собственный SSRC-идентификатор в CSRC-список для данного пакета.

Например, запись "M1:13(1,17)" обозначает пакет, отправленный смесителем MUX1, который идентифицируется случайным значением SSRC 13 и двумя CSRC-идентификаторами 1 и 17, скопированными с SSRC-идентификаторов пакетов конечных систем ES1 и ES2.

В RTP-сессии могут быть задействованы несколько смесителей и трансляторов.

Если два смесителя включены последовательно, как MUX2 и MUX3, то пакеты, полученные смесителем, могут быть уже объединены и включать CSRC-список со многими идентификаторами.

Смеситель (MUX3) должен формировать CSRC-список для исходящих пакетов, используя CSRC-идентификаторы уже смешанных входных пакетов (выход MUX2) и SSRC-идентификаторы несмешанных входных пакетов, поступивших от ES9 (E:36).

Это отмечено на рисунке для выходных пакетов смесителя MUX3 как M3:99(9,11,36).

Информационные RTP-пакеты не имеют поля длины или каких-либо других средств ограничения размеров пакета, по этой причине RTP полагается на нижележащий протокол при задании размера поля данных.

Максимальная длина RTP-пакетов ограничена размером используемых транспортных пакетов (например, UDP).