Бессонов Л.В., Брагина И.Г.

Операционные системы. Компьютерные сети.

Пособие для студентов, обучающихся по дополнительной квалификации «Компьютерная графика и веб-дизайн»

2009

УДК 004.7(072.8)
ББК 32.973.202я73
Б53

Бессонов Л.В., Брагина И.Г.

Б53 Операционные системы. Компьютерные сети: Пособие для студентов, обучающихся по дополнительной квалификации «Компьютерная графика и веб-дизайн» — Саратов: Изд-во «Научная книга», 2009. — 40 с.

ISBN 978-5-9758-1063-2

Данное пособие представляет собой методическую разработку в поддержку курса «Операционные системы, компьютерные сети, Internet» по дополнительной квалификации «Компьютерная графика и веб-дизайн».

В первой части пособия изложены общие теоретические материалы по устройству компьютерных сетей, служб и сервисов Internet. Вторая и третья главы содержат подробные инструкции по установке и настройке веб-сервера Apache, дополнений к нему. Подробно рассмотрены особенности установки и настройки компонентов в различных операционных системах.

Рецензенты:

Кафедра прикладной информатики
Саратовского государственного университета им.Н.Г. Чернышевского
Зав. кафедрой, К.ф.-м.н., доцент Шевырёв С.П.

УДК 004.7(072.8)
ББК 32.973.202я73

Работа издана в авторской редакции

ISBN 978-5-9758-1063-2

Глава 1. Структура и сервисы Интернет

Кратко о компьютерных сетях

В настоящее время существует немало различных технологий передачи данных. С каждым годом появляются новые технические средства, позволяющие всё быстрее и надёжнее передавать данные (например, в настоящее время бурно развиваются технологии оптической передачи данных). Ещё более бурно развиваются прикладные технологии сервисы.

Для того чтобы всё это многообразие технологий функционировало как единое целое, изначально развитие всех этих технологий подчинили общему стандарту — ISO/OSI.

Эталонная модель OSI, иногда называемая стеком OSI описывает взаимодействие возможно совершенно различных по устройству открытых систем. Модель представляет собой 7-уровневую сетевую иерархию разработанную Международной организацией по стандартам (International Standardization Organization — ISO). Эта модель содержит в себе по сути 2 различных модели:

1. горизонтальную модель на базе протоколов, обеспечивающую механизм взаимодействия программ и процессов на различных машинах
2. вертикальную модель на основе услуг, обеспечиваемых соседними уровнями друг другу на одной машине

В данном контексте протоколом будем называть договорённость о правилах взаимодействия.

В горизонтальной модели двум программам требуется общий протокол для обмена данными. В вертикальной — соседние уровни обмениваются данными с использованием интерфейсов API.

Уровень 1, физический

Физический уровень получает пакеты данных от вышележащего канального уровня и преобразует их в оптические или электрические сигналы, соответствующие 0 и 1 бинарного потока. Эти сигналы посылаются через среду передачи на приемный узел. Механические и

электрические/оптические свойства среды передачи определяются на физическом уровне и включаются:

- Тип кабелей и разъемов
- Разводку контактов в разъемах
- Схему кодирования сигналов для значений 0 и 1

К числу наиболее распространенных спецификаций физического уровня относятся:

- EIA-RS-232-C, CCITT V.24/V.28 — механические/электрические характеристики несбалансированного последовательного интерфейса.
- EIA-RS-422/449, CCITT V.10 — механические, электрические и оптические характеристики сбалансированного последовательного интерфейса.
- IEEE 802.3 — Ethernet
- IEEE 802.5 — Token ring

Уровень 2, канальный

Канальный уровень обеспечивает создание, передачу и прием кадров данных. Этот уровень обслуживает запросы сетевого уровня и использует сервис физического уровня для приема и передачи пакетов. Спецификации IEEE 802.x делят канальный уровень на два подуровня: управление логическим каналом (LLC) и управление доступом к среде (MAC). LLC обеспечивает обслуживание сетевого уровня, а подуровень MAC регулирует доступ к разделяемой физической среде.

Наиболее часто используемые на уровне 2 протоколы включают:

- HDLC для последовательных соединений
- IEEE 802.2 LLC (тип I и тип II) обеспечивают MAC для сред 802.x
- Ethernet
- Token ring
- FDDI
- X.25
- Frame relay

Уровень 3, сетевой

Сетевой уровень отвечает за деление пользователей на группы. На этом уровне происходит маршрутизация пакетов на основе преобразования

MAC-адресов в сетевые адреса. Сетевой уровень обеспечивает также прозрачную передачу пакетов на транспортный уровень.

Наиболее часто на сетевом уровне используются протоколы:

- IP - протокол Internet
- IPX - протокол межсетевого обмена
- X.25 (частично этот протокол реализован на уровне 2)
- CLNP - сетевой протокол без организации соединений

Уровень 4, транспортный

Транспортный уровень делит потоки информации на достаточно малые фрагменты (пакеты) для передачи их на сетевой уровень.

Наиболее распространенные протоколы транспортного уровня включают:

- TCP - протокол управления передачей
- NCP - Netware Core Protocol
- SPX - упорядоченный обмен пакетами
- TP4 - протокол передачи класса 4

Уровень 5, сеансовый

Сеансовый уровень отвечает за организацию сеансов обмена данными между оконечными машинами. Протоколы сеансового уровня обычно являются составной частью функций трех верхних уровней модели.

Уровень 6, уровень представления

Уровень представления отвечает за возможность диалога между приложениями на разных машинах. Этот уровень обеспечивает преобразование данных (кодирование, компрессия и т.п.) прикладного уровня в поток информации для транспортного уровня. Протоколы уровня представления обычно являются составной частью функций трех верхних уровней модели.

Уровень 7, прикладной

Прикладной уровень отвечает за доступ приложений в сеть. Задачами этого уровня является перенос файлов, обмен почтовыми сообщениями и управление сетью.

К числу наиболее распространенных протоколов верхних уровней относятся:

- FTP — протокол переноса файлов
- TFTP — упрощенный протокол переноса файлов
- X.400 — электронная почта
- Telnet — протокол удалённого управления
- SMTP — простой протокол почтового обмена
- CMIP — общий протокол управления информацией
- SNMP — простой протокол управления сетью
- NFS — сетевая файловая система
- FTAM — метод доступа для переноса файлов

Кратко о семействе протоколов TCP/IP

Сетевые протоколы создаются на основе концептуальной модели взаимодействия открытых систем OSI/ISO, в которой каждый уровень отвечает за свою часть процессов передачи информации.

- *Семейством протоколов (protocol suite)* называют всю совокупность протоколов различных уровней.

Семейство TCP/IP принято разделять на четыре уровня:

1. *Канальный уровень*[1] (link layer, data-link layer), или уровень сетевого интерфейса (network interface). Состоит из двух основных компонент: аппаратного сетевого интерфейса и соответствующего драйвера этого сетевого интерфейса в операционной системе. Вместе они обеспечивают физическое подключение к кабелю (или иной среде передачи), так и управление всеми аппаратными процессами передачи.
2. *Сетевой уровень (network layer, internet layer)* отвечает за перемещение пакетов по тому или иному маршруту в сети. В семействе протоколов TCP/IP сетевой уровень представлен протоколами IP (Internet protocol), ICMP (Internet Control Message Protocol) и IGMP (Internet Group Management Protocol).

[1] В русскоязычной литературе *канальный уровень* часто носит различные названия. К примеру, его было принято называть уровнем звена передачи данных, расположив под ним самостоятельный *физический уровень*. В семействе TCP/IP эти два уровня едины.

3. *Транспортный уровень (transport layer)* организует для вышестоящего прикладного уровня обмен данными между двумя компьютерами в сети. В семействе TCP/IP одновременно используются два существенно различных транспортных протокола: TCP (Transmission Control Protocol — протокол управления данных) и UDP (User Datagram Protocol — протокол дейтаграмм пользователя).

TCP обеспечивает надёжную передачу потоков данных между двумя компьютерами в сети. В его задачи входит:

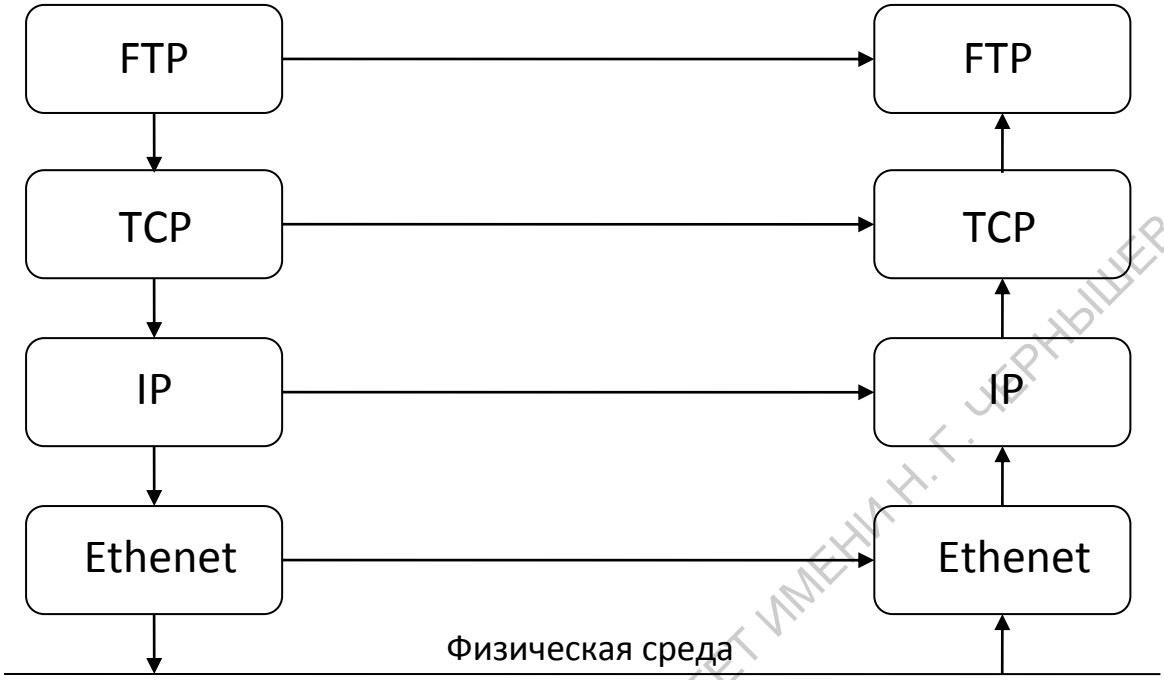
- разделять данные, поступающие от обслуживаемых им приложений, на блоки приемлемого размера для нижестоящего сетевого уровня;
- подтверждать получение пришедших к нему пакетов;
- в течение установленных им периодов времени ожидать прихода подтверждений получения отправленных им пакетов и др.

Поскольку TCP берёт на себя все задачи обеспечения надёжной доставки переданных ему данных до места назначения, то прикладной уровень освобождается от этих задач.

UDP предоставляет прикладному уровню существенно более простой сервис. Он рассылает данные адресатам в виде пакетов, называемых UDP-дейтаграммами (UPD datagrams), без гарантий их доставки. Предполагается что требуемая степень надежности пересылки должна обеспечиваться самим прикладным уровнем.

4. *Прикладной уровень (application layer)* обеспечивает выполнение различных прикладных задач. Существует определённый классический набор прикладных сервисов, которые предполагаются в большинстве реализаций семейства TCP/IP. В их числе: Telnet (протокол удалённого доступа), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), SNMP (Simple Network Management Protocol).

Пример 0.1. Взаимодействие двух компьютеров одной локальной сети (обмен файлами по FTP)



На прикладном уровне два взаимодействующих объекта: FTP-клиент и FTP-сервер. Такое распределение ролей характерно для большинства сетевых приложений, а соответствующий концептуальный подход в проектирование называется *архитектурой клиент-сервер*. В общем случае сервер — это та сторона, которая предоставляет партнёру определённые услуги.

На каждом уровне реализован один (или более) протокол, служащий для дистанционного общения с соответствующим *протоколом-партнёром (peer)* на удалённой рабочей станции. В нашем случае транспортные уровни двух рабочих станций взаимодействуют по протоколу TCP, а их сетевые уровни — по протоколу IP.

Существует принципиальное различие между верхним и тремя нижними уровнями. Прикладной уровень отвечает лишь за выполнение самой прикладной задачи, а не за передачу данных по сети. Три нижних уровня, наоборот, не имеют отношения к прикладным аспектам задачи, но в совокупности они перекрывают весь спектр проблем, связанных с организацией обмена данными.

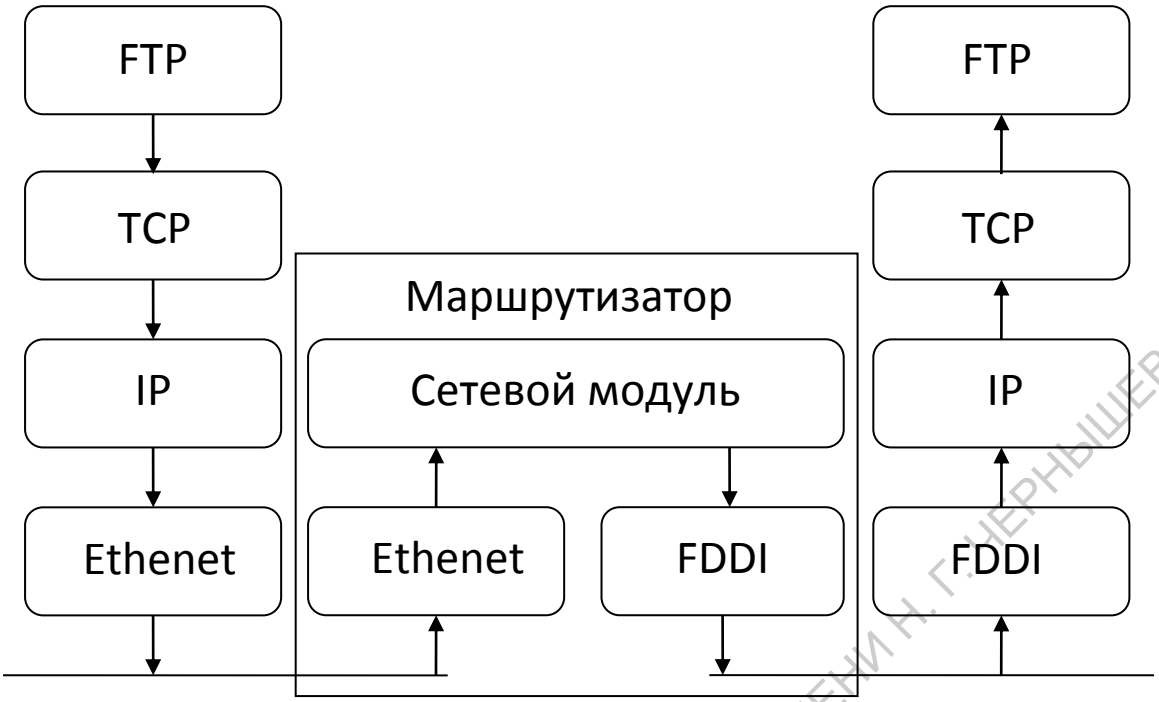
На рисунке мы видим четыре протокола: FTP — на прикладном уровне, TCP — на транспортном, IP — на сетевом, а протокол локальной сети Ethernet — на канальном. Семейство протоколов TCP/IP объединяет множество протоколов, из которых лишь два отражены в его общепринятом названии.

Назначение нижнего и верхнего уровней достаточно очевидно: канальный уровень занимается процессами передачи данных по присоединённой физической сети, а прикладной уровень обеспечивает выполнение отдельных задач пользователя. Что касается различия между сетевым и транспортным уровнем, то оно не очевидно на первый взгляд. Чтобы понять причину имеющего место быть расслоения рассмотрим взаимодействие нескольких сетей.

Для объединения двух и более сетей обычно используют маршрутизаторы. Как правило, это специализированные устройства, способные объединять локальные сети самых различных типов (Ethernet, Token Ring, FDDI и т.п.), а также связывать их по каналам точка-точка (point-to-point links).

Замечание 0.1

В современной литературе используют название *IP-маршрутизатор* (IP router) или, учитывая контекст, просто *маршрутизатор* (router). Исторически сложилось что в США для этих устройств применялось название gateway, и оно до сих пор встречается в литературе. Однако теперь терминами gateway и gate чаще обозначают *шлюзы*. Шлюз одновременно поддерживает два различных стека протоколов (например TCP/IP и SNA фирмы IBM), позволяя взаимодействовать функционально сходным приложениям, реализованным в стандартах разных семейств. Шлюзы чаще всего бывают «проблемно-ориентированными» и применяются для конкретных задач — например, для «стыковки» двух различных систем электронной почты или файлового обмена



На рисунке изображена интерсеть, состоящая из двух локальных сетей (Ethernet и FDDI), объединённых маршрутизатором. Для простоты в каждой сети показано по одному компьютеру, хотя, конечно, любая рабочая станция может связываться через маршрутизатор с любой другой станцией, находящейся в сопредельной сети.

Рисунок позволяет понять различие между двумя типами узлов в сети: *оконечные системы (end systems)* — это взаимодействующие рабочие станции, а *промежуточные системы (intermediate systems)* — это маршрутизаторы. Прикладной уровень и транспортный уровень используют *сквозные оконечные протоколы (end-to-end protocols)*. Как видно из рисунка эти два уровня присутствуют только на оконечных системах. Сетевой же уровень обеспечивает протокол *поэтапной передачи (hop-by-hop protocol)*. Он применяется как на всех оконечных системах, так и на каждом промежуточном узле.

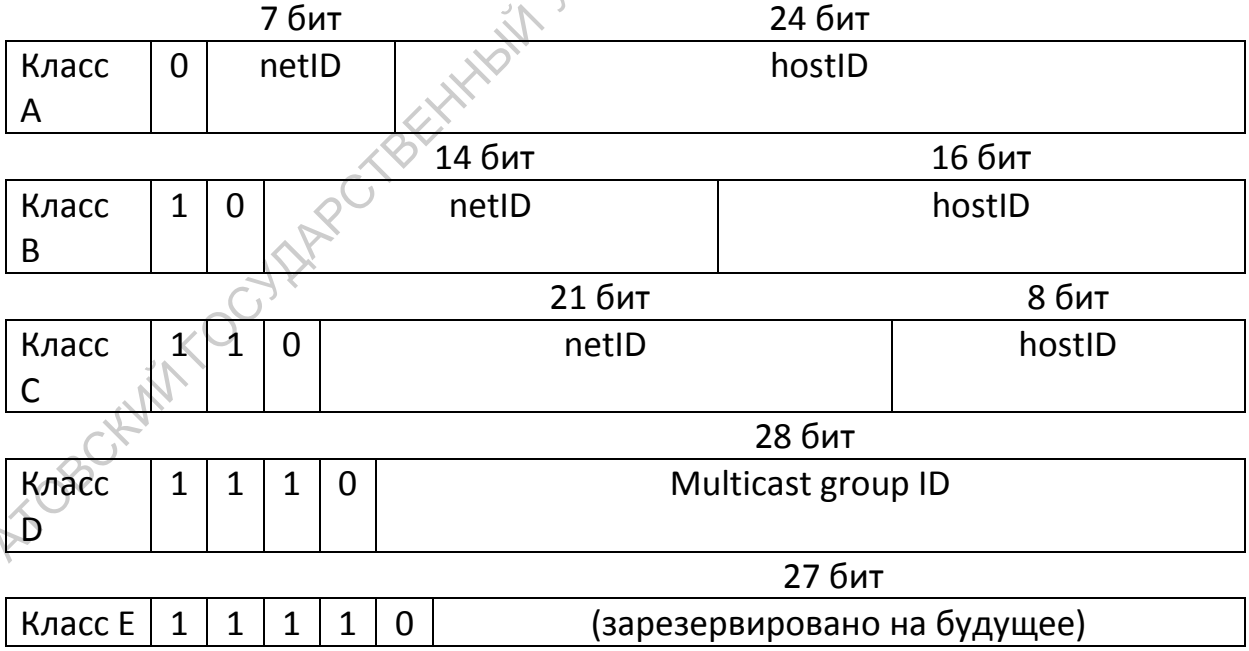
В семействе протоколов TCP/IP сетевой уровень (IP-уровень) предоставляет ненадёжный сервис. Другими словами, он направляет пакет из исходного пункта к пункту назначения, но не может гарантировать успешную доставку. Напротив, TCP, имея в своём распоряжении лишь ненадёжную сетевую службу IP, тем не менее обеспечивает приложениям надёжный транспортный сервис. Для этого TCP выполняет большой объём

дополнительной работы: он подтверждает приём пакетов и принимает подтверждения от своего партнёра при обмене, устанавливает таймауты ожидания подтверждений на свои пакеты и, если подтверждения не приходят, производит *повторную передачу данных (retransmission)* и т.п. Отсюда видим что функции транспортного и сетевого уровня существенно дополняют друг друга.

Другой способ объединения локальных сетей — использование мостов (bridges). В отличие от маршрутизаторов, действующих сетевом уровне, мосты объединяют сети на канальном уровне. В результате связанные мостами локальные сети выглядят топологически как одна сеть. Аналогично, с *коммуникаторами*.

Адресация в сети Интернет

Каждый сетевой интерфейс в Internet должен иметь свой уникальный IP-адрес. IP-адрес содержит 32 двоичных разряда (4 байта). Множество IP-адресов, образующих адресное пространство, структурировано: оно разбито на 5 различных классов.



netID — идентификатор сети
hostID — идентификатор хоста
multicast group ID — идентификатор группы (групповой адрес)

Конкретные IP-адреса принято записывать десятичными значениями (0..255) четырёх байтов, разделённых точками. Такой способ записи называют точечной десятичной записью.

Распознать класс сети в заданном адресе легко — он определяется первым числом в точечно-десятичной записи.

Класс	Диапазон адресов
A	0.0.0.0 — 127.255.255.255
B	128.0.0.0 — 191.255.255.255
C	192.0.0.0 — 223.255.255.255
D	224.0.0.0 — 239.255.255.255
E	240.0.0.0 — 247.255.255.255

Так как всякий интерфейс должен иметь свой уникальный IP-адрес, то необходима централизованная служба, распределяющая адреса для сетей, присоединяемых к всемирной сети Internet. Такой службой является InterNIC (Internet Network Information Center — сетевой информационной центр Internet). InterNIC занимается регистрацией сетей, предоставляет каждой уникальный идентификатор (netID). Распределение конкретных IP-адресов в данной сети между её хостами — задача, возлагаемая на системного администратора сети.

По числу адресатов различают три типа IP-адресов: *однозначный* (unicast) — предназначенный для единственного хоста, *широковещательный* (broadcast) — для всех хостов в одной сети и *групповой* (multicast) — для нескольких хостов, образующих группу рассылки.

Со второй половины 90-х годов XX века классовая адресация повсеместно вытеснена бесклассовой адресацией, при которой количество адресов в сети определяется только и исключительно маской подсети.

Особые IP-адреса

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

- если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет; этот режим используется только в некоторых сообщениях ICMP;

- если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет; этот режим используется только в некоторых сообщениях ICMP;
- если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (*limited broadcast*);
- если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, в сети *192.190.21.0* с маской *255.255.255.0* пакет с адресом *192.190.21.255* доставляется всем узлам этой сети. Такая рассылка называется широковещательным сообщением (*broadcast*).

Кроме того существуют некоторые зарезервированные адреса и диапазоны.

127.0.0.1—127.255.255.255 зарезервированный диапазон IP-адресов для обозначения так называемого «локального хоста», то есть для сети, состоящей только из одного компьютера. Как правило, используется всего один адрес — 127.0.0.1, который устанавливается на специальный сетевой интерфейс «внутренней петли» («loopback») в сетевом протоколе TCP/IP. В Unix-подобных системах данный интерфейс обычно именуется «loX», где X — число, либо просто «lo». При установке соединений в этой вырожденной «сети» присутствует только один компьютер («Я»), при этом сетевые протоколы выполняют функции протоколов межпроцессного взаимодействия.

Использование адреса 127.0.0.1 позволяет устанавливать соединение и передавать информацию для программ-серверов, работающим на том же компьютере, что и программа-клиент, независимо от конфигурации аппаратных сетевых средств компьютера (не требуется сетевая карта, модем, и прочее коммуникационное оборудование, интерфейс реализуется при помощи драйвера псевдоустройства в ядре операционной системы). Таким образом, для работы клиент-серверных

приложений на одном компьютере не требуется изобретать дополнительные протоколы и дописывать программные модули.

Обычно адресу 127.0.0.1 однозначно сопоставляется имя хоста «localhost» и/или «localhost.localdomain».

Инкапсуляция данных

Когда прикладная программа посылает данные при помощи протокола TCP (UDP), то они проходят через все уровни вниз по протокольному стеку, прежде чем превратятся в поток битов, передаваемых в линию. Каждый слой добавляет к спущенной ему порции данных собственную информацию в виде своего заголовка, а иногда и концевика.

46—1500 байтов

Ограничения на длину кадра в локальной сети Ethernet обусловлены физическими особенностями: размер поля данных должен быть в пределах от 46 до 1500 байтов.

Разные приложения могут одновременно передавать данные как по TCP, так и по UPD, при этом необходимо соотносить передаваемые данные приложениями. С этой целью TCP и UDP используют специальные 16-разраядные *номера портов*. Номера порта источника и порта назначения записываются в соответствующих полях заголовков TCP (UDP).

Демультиплексирование данных

Демультиплексирование данных (demultiplexing) — это многоступенчатый процесс обработки принятых данных, охватывающий все уровни модели ISO/OSI, и подразумевающий, что с момента получения хостом

адресованного ему кадра Ethernet содержащиеся в этом кадре данные начинают продвигаться по стеку протоколов вверх, последовательно освобождаясь от обрамляющих их заголовков по мере того, как эти заголовки обрабатываются соответствующими протокольными модулями. Каждый протокольный модуль анализирует определённые информационные поля в соотнесённом ему заголовке (см. схему инкапсуляции), чтобы узнать какому из вышестоящих модулей передать вложенные данные.

Архитектура клиент-сервер

Большинство сетевых приложений реализованы по технологии «клиент-сервер». Это означает, что программное обеспечение делится на две части: клиентскую, устанавливаемую на компьютере пользователя, и серверную, устанавливаемую на компьютере-сервере. Прикладная задача выполняется в результате того, что сервер определенным образом обслуживает запросы клиента.

Клиентское программное обеспечение предназначено для формирования запросов пользователя к серверной части и для отображения пользователю получаемых от сервера ответов. Клиентская часть обеспечивает удобство пользовательского интерфейса. Основную же нагрузку несет серверная часть, которая обычно обслуживает запросы множества клиентов.

Программы, реализующие различные службы в Интернете, относятся к серверам. Программы, с помощью которых пользователь получает доступ к этим службам, относятся к категории клиентов.

Службы и сервисы

По всему вышеописанному «привычный нам Интернет» трудно узнаётся. Напрашивающийся вывод: Интернет — это компьютерная сеть, то есть совокупность компьютеров и коммуникационных средств.

Рядовой пользователь привык воспринимать Интернет иначе. Первые ассоциации пользователя со словом «Интернет»: сайты, почта и т.п. Всё это реализуется при помощи служб и сервисов Интернета.

Интернет с самого начала обладал открытой архитектурой. Это означает, что новые службы могут возникать по мере необходимости. Некоторые из них становятся популярными и процветают, некоторые отвечают потребностям ограниченного круга пользователей, некоторые вытесняются более совершенными конкурирующими службами. Все зависит от потребностей людей в данном способе обмена информацией и, в какой-то степени, от моды и от привычки.

В телефонии каждая новая служба означает новое устройство, как, например, факсимильный аппарат. Однако компьютеры — это универсальные устройства, и TCP/IP предоставляет им универсальное средство связи. Поэтому в Интернете новая служба — это просто другая программа. Почти любой программист может создать целую индустрию, придумав новую службу для Интернета.

Наиболее широко используются следующие службы Интернета:

- *telnet, ssh, remote desktop* — позволяют вам соединиться с удаленным компьютером и работать с ним так, как будто вы сидите перед ним, в текстовом или графическом режиме. Это то, для чего предназначался Интернет в момент его зарождения. Теперь эта служба используется прежде всего теми, кто следит за бесперебойной работой сети, — системными администраторами. Чаще используют службу, которая шифрует передаваемую информацию, - *ssh*.
- *ftp* - также одна из старейших служб, используется для копирования файлов с компьютера на компьютер. В ftp-архивах Интернета можно найти много полезных программ.
- *e-mail* (электронная почта) — в соответствии с названием, почта, только электронная. Выполняет те же функции, что и обычная почта, только быстрее, надежнее и дешевле. Это самая главная служба в Интернете на протяжении 80-х годов, и она ничуть не потеряла своего значения сейчас. Вы можете не пользоваться никакими другими службами Интернета, но этой пользуются все.
- *WWW* (веб) — служба, которая совмещает в Интернете функции электронного издательства и библиотеки. Особенность публикаций в Интернете — это широкое использование ссылок и отсутствие

разницы для читателя между ссылками внутри документа и ссылками на другие документы, где бы они ни хранились. С точки зрения читателя все публикации в Интернете представляют собой один постоянно дописываемый многими авторами гигантский документ, связанный паутиной перекрестных ссылок, что и дало название этой службе (Всемирная Паутина). Эта служба появилась в начале 90-х годов и стала невероятно популярной.

Следует отметить что большинство пользователей отождествляют Интернет с одной единственной его службой — WWW. Такой взгляд ошибочен.

Служб существует огромное количество, многие из них, появившись на заре Интернета, получают новую жизнь с появлением новых технологий. Так, например, с появлением субпротокола SSL, позволяющего организовать шифрованную передачу данных, модификацию претерпели старейшие службы интернета, появились службы ftps, https, целый ряд почтовых служб, включающих SSL, и т.д.

Службы реализуются протоколами прикладного уровня. Например, на компьютере могут быть установлены FTP-клиент, HTTP-клиент (веб-браузер) и прочие клиенты. Все они будут функционировать на верхнем уровне стека TCP/IP.

Данные поступающие различным клиентам и серверам, установленным на компьютере, не смешиваются между собой благодаря процессу демультиплексирования. В этом процессе все данные сортируются по *портам.* По сути, порт — это некий уникальный номер. Деление потока данных по портам реализовано на транспортному уровне, то есть протоколами TCP и UDP. Например, существует договорённость о том, что веб-сервер работает на порту номер 80. Следуя этой договорённости, когда пользователь пишет в веб-браузере свой запрос «http://www.sgu.ru/», браузер узнаёт IP-адрес узла `www` сети `sgu.ru` и направляет запрос на получение информации на 80 порт этого узла.

Существует набор общепринятых договорённостей о нумерации портов — well-known port numbers (WKPN). Благодаря этой договорённости известно

на каких номерах портов следует запрашивать (если не указан номер порта) стандартные службы.

В операционной системе Windows можно посмотреть на зарезервированные номера портов стандартных службы в файле `services`, находящемся в каталоге `System32\drivers\etc`. Это текстовый файл, который можно открыть блокнотом.

Система именования доменов (DNS)

Сервер имен это программа управления распределенной базой данных, в которой хранятся символьные имена сетей и хостов вместе с их IP-адресами. Наиболее распространенным программным продуктом, решающим данную задачу является BIND (Berkrley Internet Name Domain). Иногда для этой цели выделяют специальную машину. Задача DNS — преобразование символьного имени в IP-адрес и наоборот в условиях, когда число узлов Internet растет экспоненциально, совсем не проста. Сама иерархическая система имен (DNS) настроена на упрощение решения этой проблемы. Схема взаимодействия программы пользователя с локальным и удаленными DNS-серверами показана ниже на рисунке 1.

Рис. 1. Структура взаимодействия с серверами имен

База имен является распределенной, так как нет такой компьютера, где бы хранилась вся эта информация. Имя содержит несколько полей (длиной не более 63 символов каждое), разделенных точками. Целиком имя может содержать не более 255 октетов, включая байт длины. Анализ имени производится справа налево. Самая правая секция имени характеризует страну (двухсимвольные национальные коды), или характер организации (образовательная, коммерческая, правительственная и т.д.).

Следующие 3-х символьные коды в конце Internet-адреса означают функциональную принадлежность узла:

Таблица 1. Стандартизованные TLD-суффиксы имен

Поле адреса	Тип сети
.aero	Фирма или организация, относящаяся к сфере авиации;
.biz	Организация, относящаяся к сфере бизнеса;
.com	Коммерческая организация;
.coop	Кооперативная организация;
.gov	Государственное учреждение (США);
.info	Открытая TLD-структура (регистрация имен доменов)
.org	Бесприбыльная организация;
.edu	Учебное заведение;
.mil	Военное предприятие или организация (США);
.museum	Имя домена музея
.name	Имя домена частного лица
.net	Большая сеть;
.pro	Профессионал, достойный доверия. Управляется RegistryPro (http://www.nic.pro/);
.int	Международная организация;
.arpa	Специальный домен, используемый для преобразования ip-адреса в имя

Секции .mil и .gov принадлежат исключительно американским сетям (хотя и многие другие трехсимвольные секции адреса, например .edu, чаще, но не всегда, принадлежат американским университетам и другим учебным организациям). Структура имен обычно отражает структуру организации, которой принадлежат сети или хосты, но не архитектуру сети в этой организации. Так имя sgu.ru — это имя домена sgu (Саратовский Государственный университет). mexmat.sgu.ru — имя конкретного хоста, immpu.sgu.ru — имя подсети (Институт механики, математики и процессов управления). По имени, как правило, нельзя определить является ли оно

именем сети, маршрутизатора или конкретного хоста. Для записи символьных имен используется исключительно латинский алфавит.

Маленький фрагмент интернетовской иерархии имен показан на рис. 2. Число уровней реально больше, но обычно не превышает 5.

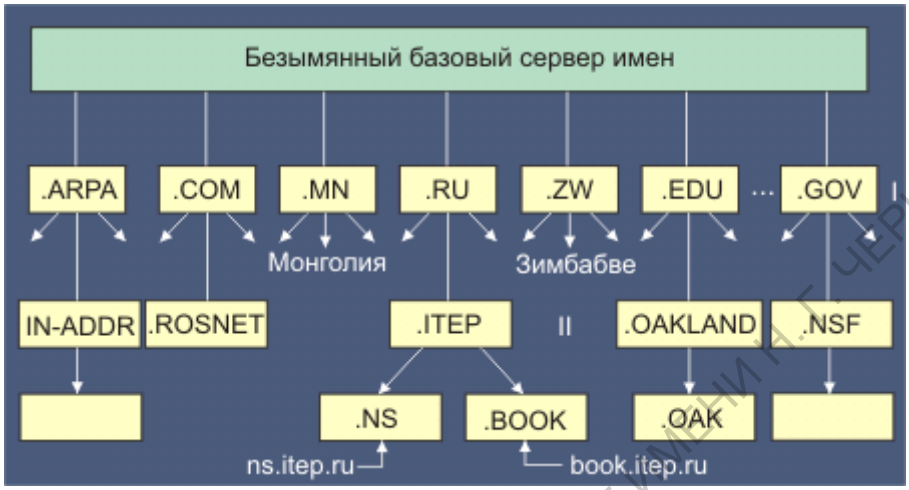


Рис. 2. Иерархия имен в Интернет-DNS (I - домен первого уровня; II - второго уровня)

Каждому узлу (прямоугольнику на рисунке) соответствует имя, которое может содержать до 63 символов. Только самый верхний, корневой узел не имеет имени. При написании имени узла строчные и прописные символы эквивалентны. Имя домена, завершающееся точкой называется абсолютным или полным именем домена (например, itep.ru.). В некоторых странах создана структура имен сходная с com/edu/org. Так в Англии можно встретить адреса .ac.uk для академических учреждений и .co.uk — для коммерческих. Если имя домена не завершено символом точки, DNS может попытаться его дополнить, например имя ns может быть преобразовано в ns.sgu.ru. На приведенной схеме каждому объекту трех верхних уровней соответствуют серверы имен, которые могут взаимодействовать друг с другом при решении задачи преобразования имени в IP-адрес. Можно было бы построить схему, при которой в любом сервере имен имелись адреса серверов .com, .edu, .ru и т.д. и при необходимости он мог бы послать туда запрос. Реальная схема серверов не столь идеальна и стройна — существуют серверы nsf.gov, oakland.edu и т.д., которые непосредственно связаны с базовым сервером имен.

Каждый сервер содержит лишь часть дерева имен. Эта часть называется зоной ответственности сервера. DNS-сервер может делегировать ответственность за часть зоны другим серверам, создавая субзоны. Когда в зоне появляется новый хост или субдомен, администратор зоны записывает ее имя и IP-адреса в базу данных сервера. Администратор зоны определяет, какой из DNS-серверов имен является для данной зоны первичным. Число вторичных серверов не лимитировано. Первичный и вторичный серверы должны быть независимыми и работать на разных ЭВМ, так чтобы отказ одного из серверов не выводил из строя систему в целом. Отличие первичного сервера имен от вторичного заключается в том, что первичный загружает информацию о зоне из файлов на диске, а вторичный получает ее от первичного. Администратор вносит любые изменения в соответствующие файлы первичного сервера, а вторичные серверы получают эту информацию, периодически (раз в 3 часа) запрашивая первичный сервер. Пересылка информации из первичного во вторичные серверы имен называется зонным обменом. Как будет вести себя сервер, если он не имеет запрашиваемой информации? Он должен взаимодействовать с корневыми серверами. Таких серверов насчитывается около десяти и их IP-адреса должны содержаться в конфигурационных файлах.

Список корневых серверов вы можете получить с помощью анонимного FTP по адресу ftp.rs.internic.net, конкретно текущую версию корневой зоны можно получить по адресу ftp://ftp.internic.net/domain/named.root. Корневые серверы хранят информацию об именах и адресах всех серверов доменов второго уровня. Существует два вида запросов: рекурсивные и итеративные. Первый вид предполагает получение клиентом IP-адреса, а второй - адреса сервера, который может сообщить адрес. Первый вид медленнее, но дает сразу IP-адрес, второй эффективнее — в вашем сервере копится информация об адресах серверов имен. Одним из способов повышения эффективности трансляции имен в адреса является кэширование, то есть хранение в оперативной памяти имен-адресов, которые использовались последнее время особенно часто. Рассмотрение процесса обмена между хостом-клиентом и сервером имен может прояснить работу системы в целом. Прежде чем использовать протоколы UDP или TCP прикладная программа должна

узнать IP-адрес объекта, куда она хочет послать дейтаграмму. Для решения этой проблемы программа посылает запрос в локальный сервер имен. В Unix-системах имеются специальные библиотечные процедуры gethostbyname и gethostbyaddr, которые позволяют определить IP-адрес по имени ЭВМ и наоборот. В одном запросе может содержаться несколько вопросов. Если сервер не сможет ответить на вопросы, он пришлет отклик, где содержатся адреса других серверов, способных решить эту задачу.

В качестве транспорта для службы DNS используется UDP или TCP, порт 53.

Предопределение DNS

Операционные системы зачастую содержат в пакете средств взаимодействия с сетью специальную программу — DNS-клиент. В свою очередь, наиболее распространённые DNS-клиенты имеют системы *предопределения*. Эта система представляет собой локальную базу (чаще всего это обычный текстовый файл), в которой производится поиск запрошенного адреса или имени перед тем как будет сформулирован и отправлен запрос к DNS-серверу.

Эта система необычайно удобна для веб-разработчиков. Поскольку позволяет локально работать над сайтом, не регистрируя имён в системе DNS.

В операционной системе Windows система предопределения DNS также присутствует. База данных этой системы — текстовый файл hosts. Находится этот файл в папке `System32\drivers\etc`.

Формат файла предельно прост — в каждой строчке содержится запись вида:

```
IP-адрес              имя
```

Например:

```
127.0.0.1             www.mysite.ru
127.0.0.1             www.site.ru
```

Строки, начинающиеся со знака решётки «#», считаются комментариями и игнорируются системой.

В операционной системе Linux DNS-клиенты как правило обладают такой же системой, а файл hosts находится в каталоге `/etc`.

Кибersквоттинг

Кибersквóттинг (англ. *cybersquatting*) — приобретение доменных имён, созвучных названиям известных компаний, или просто с «дорогими» названиями с целью их дальнейшей перепродажи или размещения рекламы. Люди, практикующие такие действия, называются *кибersквóттерами*.

Виды кибersквоттинга

Можно выделить следующие виды занятий, обычно объединяемых термином «кибersквоттинг» или «захват доменов».

- Тайпсквоттинг — регистрация доменных имён, близких по написанию с адресами популярных сайтов в расчёте на ошибку части пользователей. Например, «wwwsite.ru» в расчёте на пользователя, который хотел попасть на «www.site.ru». При близости к очень популярным доменам тайпскворттер может собрать на своём сайте некоторый процент «промахнувшихся» посетителей и засчёт показа рекламы заработать денег. Попробуйте набрать адрес *microsoft.com* с любой опечаткой, проверьте существует «адрес с опечаткой».

- Брендовый кибersквоттинг — регистрация доменных имён, содержащих товарные знаки, фирменные наименования, популярные имена собственные, то есть средства индивидуализации, охраняемые законом. Хотя при этом у кибersкворттера есть риск лишиться домена и подвергнуться ответственности, законные владельцы товарных знаков могут предпочесть не судиться, а выкупить захваченные домены.

- Защитный кибersквоттинг — когда легальный владелец популярного сайта (товарного знака) регистрирует все доменные имена, близкие, созвучные, похожие, связанные по смыслу с его собственным доменным именем. Делается для того, чтобы не стать жертвой кибersквоттеров. Например, владелец популярного сайта «www.firma.ru» может захотеть также зарегистрировать домены «firma-msk.ru», «firma-spb.ru» и «firma.org», чтобы перенаправлять с них посетителей на свой основной сайт, а также «anti-firma.ru», чтобы не использовать его.

Бывают случаи, когда доменное имя захватывают с целью шантажа или вымогательства. Кибersквоттер угрожает разместить на захваченном домене сайт с негативной информацией или выдавать этот сайт за принадлежащее правомерному владельцу средство индивидуализации. Впрочем, этот вид кибersквоттинга отличается повышенным риском уголовного преследования и применяется редко. Для иллюстрации можно вспомнить историю с сайтами «lugkov.ru» и «lujkov.ru». Другой пример — подложный сайт Генпрокуратуры «gprf.info» (ныне не существует).

Принципы действия кибersквоттеров

Кибersквоттер не может позволить себе регистрировать в качестве доменного имени все сколь-нибудь осмысленные сочетания символов. Ему надо выбирать наиболее перспективные. Чтобы предсказать, какие слова станут популярными в будущем, есть несколько методов.

Сквоттеры стараются получить доступ к статистике поисковых запросов популярных поисковых систем. Когда в статистике появляются новые слова и сочетания, либо отмечается неожиданный рост частоты каких-то слов сверх обычного, это верный признак, что очень скоро слово может появиться в популярных доменных зонах, прежде всего, com. Практика подсказывает, что имя, появившееся в com, с большой вероятностью скоро потребуется и в других зонах.

В последнее время отмечаются случаи, когда кибersквоттеры продают домены не конечным потребителям, а друг другу. Это говорит о развитости рынка доменных имён. Домены постепенно превращаются из чисто потребительского товара в средство инвестирования.

Захват доменов и торговые марки

Согласно российскому законодательству, товарный знак имеет приоритет перед доменным именем, то есть владелец товарного знака может запретить использование своего знака в доменном имени. Фактически это означает возможность отобрать домен у кибersквоттера. Отсюда возникает возможность так называемого «обратного захвата» домена, то есть, осуществление операции отбора домена через регистрацию соответствующего товарного знака. Однако на практике этого пока не осуществлялось. Тем не менее, зная о такой возможности, многие

владельцы дорогих доменных имён регистрируют соответствующие им товарные знаки.

Некоторые интересные ссылки

Порядок регистрации доменных имён в зоне RU:
http://info.nic.ru/st/62/out_72.shtml

Информация о правовом регулировании доменных имён:
http://info.nic.ru/st/13/

Глава 2. Веб-сервер Apache

Стандартной для веб связкой сервисов является: веб-сервер и система управления базами данных (СУБД). Причём веб-сервер в этой связке должен обладать средствами для связи с СУБД. Как правило, таким средством является некоторый язык программирования.

Развернём на локальном компьютере веб-сервер Apache в связке с интерпретатором языка PHP и СУБД MySQL.

Для начала необходимо получить дистрибутивы серверов Apache и MySQL, а так же PHP. Будем устанавливать и настраивать Apache версии 2, MySQL версии 4 и PHP версии 5.

Скачать Apache можно с зеркал приведённых на официальном сайте http://www.apache.org/dyn/closer.cgi. При поиске следует помнить, что Apache так же может называться httpd, по имени его демона в UNIX. На зеркалах обычно много различных файлов, например:

1. httpd-2.0.49-win32-src.zip — это архив с исходными кодами (src) для Windows (win32) Web-сервера Apache (httpd) версии 2.0.49.
2. httpd-2.0.49.tar.gz — тоже самое, но для Linux, в котором программы принято распространять в исходных кодах.
3. apache_2.0.50-win32-x86-no_ssl.exe — откомпилированный под архитектуру (x86) для Windows (win32) без поддержки SSL(no_ssl) сервер Apache (apache) версии 2.0.50, его следует скачать для установки.

Замечание 1

Бинарные коды дистрибутивов Apache для операционной системы Windows распространяются в нескольких вариантах, как с расширением *.exe, так и *.msi и имеют название вида httpd_версия_win32_*_.msi.

MSI — Microsoft Installer. Файл с таким расширением содержит файлы и сценарий установки этих файлов в систему, интерпретируемый специальной программой (собственно Installer), обычно присутствующей в операционной системе.

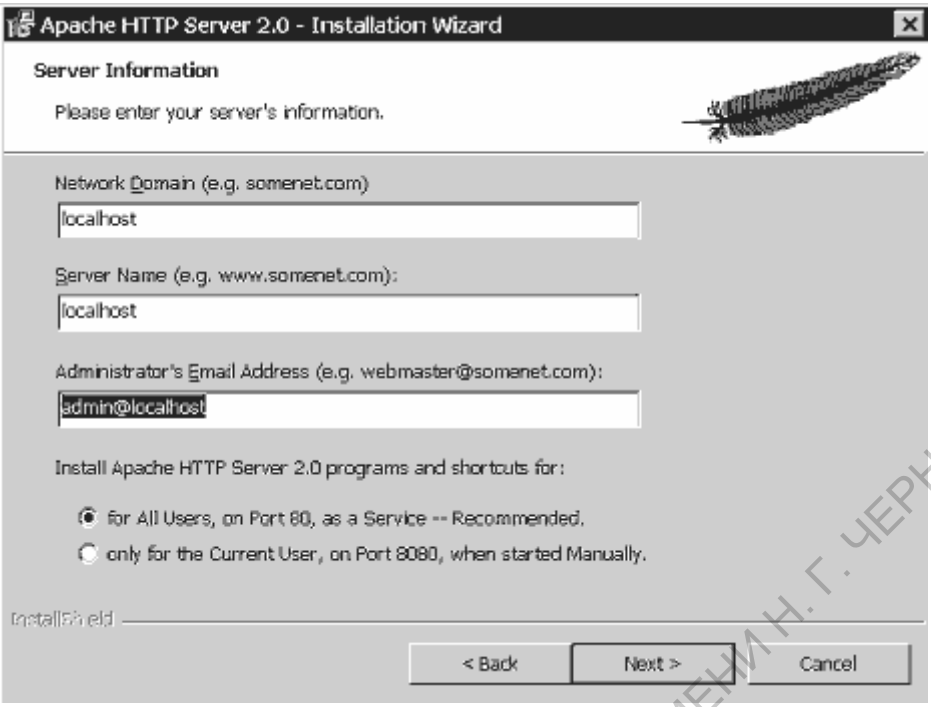
Однако возможна ситуация, когда Installer отсутствует. В этом случае необходимо либо найти эту программу и установить её в систему (например, при помощи службы обновления Windows), либо найти необходимый дистрибутив в виде исполнимого файла (exe).

Замечание 2

В случае установки Apache в операционной системе семейства *nix, воспользуйтесь стандартными средствами управления установленными пакетами. Например, для SUSE Linux это будет средство с графическим интерфейсом YaST или консольная утилита zypper (или тот же YaST для консоли).

Установка

Запустите установщик Web-сервера Apache. Результатом будет окно с лицензионным соглашением, после принятия которого, следует перейти к следующему окну с краткой информацией о нововведениях во второй версии Apache. Следующее окно, показанное на рисунке, позволяет ввести информацию о сервере: *доменное имя сервера*, *имя сервера* и *адрес электронной почты администратора*. Поскольку установка происходит на локальную машину, то в поля для доменного имени и имени сервера следует ввести *localhost* (см. рисунок.). В нижней части окна предлагается выбрать *номер порта* по которому сервер будет принимать запросы. Установим 80 (или 8080).



Напоминание

localhost — это имя для использования сервера на локальной машине, которое связано с IP-адресом 127.0.0.1, который зарезервирован для локального использования.

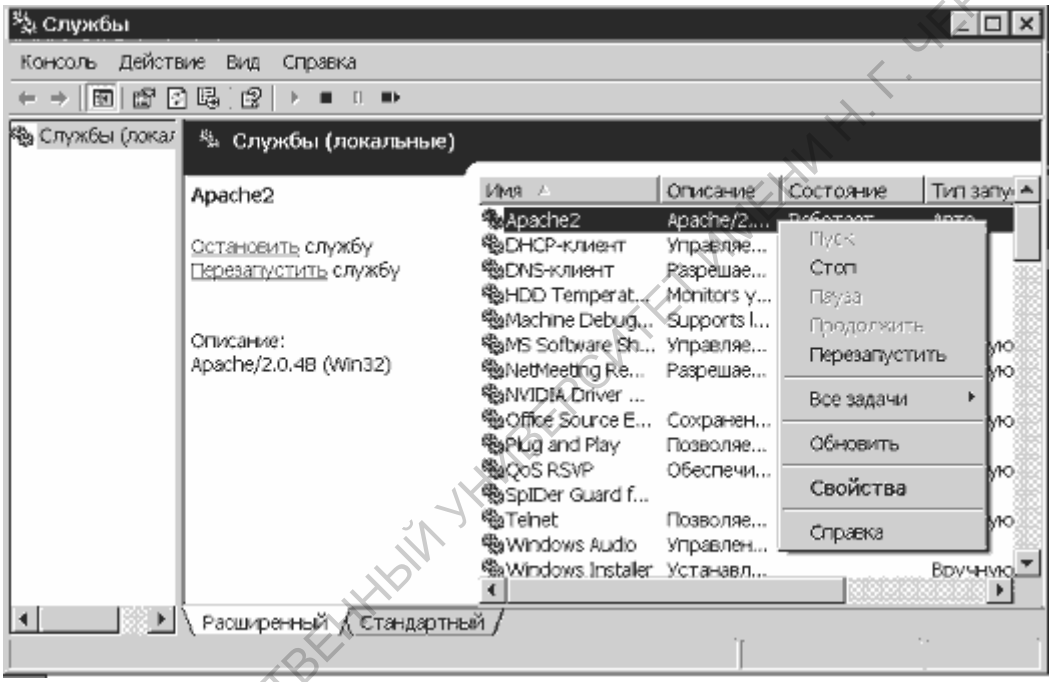
После этого будет предложен способ установки: стандартный (*Typical*) или выборочный (*Custom*), позволяющий выбрать компоненты сервера вручную. Следующее окно позволяет выбрать каталог установки сервера, по умолчанию это «`C:\Program Files\Apache Group`», рекомендуется выбрать другой каталог, например, «`C:\www`» или «`C:\Apache`». После этого мастер установки сообщит о готовности к процессу установки и после нажатия кнопки *Install*, будет произведено копирование файлов сервера. Если установка прошла успешно, Windows автоматически запустит Apache.

После успешной инсталляции при наборе в окне браузера *http://localhost/* или *http://127.0.0.1/* — должна загрузится страница сервера.

Управление службой веб-сервера

Теперь необходимо научиться управлять Apache, а именно научится запускать, останавливать и перезапускать сервер. Существует много способов осуществить эти операции: при помощи утилиты ApacheMonitor,

используя консоль управления сервисов Windows, используя пункты меню Пуск, из командной строки… Мы рассмотрим консоль управления сервисов Windows, позволяющего настроить Apache для автоматического старта при запуске системы. Для запуска консоли управления выполните команду Пуск → Настройка → Панель управления → Администрирование → Службы. В появившемся окне консоли, на приведённом ниже рисунке, следует выбрать сервис Apache2. Контекстное меню, открывающееся по нажатию на правой кнопке, позволяет осуществлять запуск, остановку и перезапуск сервиса.



Службы Windows позволяют осуществлять запуск фоновых приложений при старте системы. Для этого необходимо перейти в окно Свойства, выбрав в контекстном меню сервиса пункт *Свойства* и в появившемся окне в выпадающем списке "*Тип запуска*" выбрать пункт "*Авто*".

Конфигурирование

Web-сервер — сложный программный продукт работающий на разных платформах и в разных операционных системах по всему миру. Поэтому для корректной работы на установленной системе его необходимо настроить (сконфигурировать). По умолчанию настройки Apache расположены в файле httpd.conf в директории conf по пути установки Apache (тот путь, который был указан при установке). Далее будут описаны

основные директивы файла httpd.conf и их общеупотребительные значения.

Файл httpd.conf — главный конфигурационный файл веб-сервера Apache. Он представляет собой текстовый файл, который можно легко редактировать, например при помощи стандартного блокнота (notepad).

Условно, файл httpd.conf можно разделить на три части:

1. Конфигурация самого сервера
2. Конфигурация хоста по умолчанию
3. Конфигурация виртуальных хостов

Кроме того, в любом конфигурационном файле Apache могут присутствовать комментарии. Комментарием считается строка, начинающаяся со знака решётки «#». Содержание комментариев веб-сервером игнорируется.

Контексты конфигурации

В конфигурационном файле могут присутствовать отдельные директивы и директивы, относящиеся к некоторому контексту.

Отдельные директивы устанавливают значения глобальных параметров настройки сервера.

Контекст оформляется следующим образом:

```
<Имя_контекста цель>
# Различные инструкции
…
</Имя_контекста>
```

Некоторые часто используемые контексты:

1. Directory —параметры настройки для папки на сервере
2. Files (или FilesMatch) — параметры настройки для файла или группы файлов
3. Location — параметры настройки для расположения на сервере (отличается от Directory тем, что не обязательно должна существовать соответствующая папка)

Пути к файлам

В конфигурационных файлах Apache и PHP Вам часто придется указывать пути к различным директориям и папкам. В операционных системах UNIX и Windows применяются различные разделители каталогов. В UNIX используется прямая косая черта «/», например `/usr/bin/perl`, в Windows обратная, например, `C:\Apache`. Вообще, в некоторых директивах Apache и PHP работают оба вида разделителей каталогов: прямой «/» и обратный «\», но так как и Apache и PHP изначально разрабатывались под UNIX, то применяя их "родной" формат, Вы сможете избежать ряда проблем. Поэтому пути в настроечных файлах (httpd.conf и php.ini) рекомендуется писать через слеш в формате UNIX — «/». Например:

```
ScriptAlias "/php_dir/" "c:/php/"
```

Директивы файла httpd.conf

Port

```
Port 80
```

Устанавливает порт TCP, который используется Apache для установки соединения. По умолчанию используется 80 порт.

> *Замечание*
>
> Единственная причина использования нестандартного порта — это отсутствие прав на использование стандартного порта. При использовании нестандартного порта, например, 8080 номер порта следует указывать в адресе, например: http://localhost:8080/.

ServerAdmin

```
ServerAdmin mymail@gmail.com
```

Содержит e-mail-адрес администратора web-сервера, который будет отображаться при ошибках работы сервера.

ServerName

```
ServerName myserver
```

Содержит имя компьютера для сервера.

ServerRoot

```
ServerRoot "C:/Apache2"
```

Указывает на каталог, содержащий файлы WEB-сервера Apache.

> *Замечание*
>
> Не путайте директиву ServerRoot с директивой DocumentRoot, которая указывает каталог для файлов WEB-сайта.

DocumentRoot

```
DocumentRoot "C:/Apache2/htdocs"
```

Определяет каталог, в котором расположены файлы WEB-сайта.

DirectoryIndex

```
DirectoryIndex index.html index.phtml index.php
```

Содержит список индексных файлов, которые следует отображать при обращении к директории без указания имени файла (например, http://localhost/test/).

AddDefaultCharset

```
AddDefaultCharset windows-1251
```

Устанавливает кодировку по умолчанию, если кодировка не установлена в заголовке HTML-документа. Также Вам может потребоваться указывать значение кодировки KOI8-R или utf-8 (Unicode).

Глава 3. Установка и настройка PHP

PHP является самостоятельным интерпретатором. При желании пользователь может установить PHP и использовать его как любой другой язык программирования для написания и выполнения программ.

В текущем контексте рассмотрения PHP будет рассматриваться как язык сценариев (программ), исполняемых на стороне сервера. То есть, как приложение, задействованное в формировании ответа на запрос, полученный веб-сервером.

Схематично это выглядит так:

1. Веб-сервер получил запрос от клиента на выдачу документа, например: http://www.site.ru/news.php
2. Сервер, прочитывает этот документ и готовится отправить его клиенту, сделавшему запрос. В процессе отправки документ проходит через набор *фильтров*.
3. Фильтр определяет наличие в документе инструкций на языке php, выделяет их и передаёт интерпретатору PHP
4. Интерпретатор PHP выполняет переданные ему инструкции (например, поиск новостей в базе данных) и возвращает веб-серверу результат выполнения.
5. Фильтр заменяет в документе PHP-инструкции на возвращённые интерпретатором результаты их выполнения.
6. Документ отправляется клиенту, сделавшему запрос.

Таким образом, получается что установленный в системе PHP должен быть непосредственно связан с веб-сервером. Реализация этой связи будет подробно рассмотрена далее.

Распространяется интерпретатор PHP в двух возможных вариантах:

1. В виде установочного файла
2. В виде zip-архива, содержащего все необходимые файлы

В первом случае для установки необходимо запустить установочный файл и следовать инструкциям. В частности, наиболее важным этапом является указание установленного в системе веб-сервера и папки, в которой он

установлен. Мастер установки использует эти данные, чтобы автоматически установить взаимодействие интерпретатора PHP с веб-сервером.

Рассмотрим ситуацию «ручного режима» установки из zip-архива.

Для установки PHP следует создать каталог `c:\php` и разместить в нём файлы из zip-архива дистрибутива. После этого следует переименовать конфигурационный файл php.ini-dist в php.ini и скопировать его в директорию Windows.

Далее, необходимо сообщить Web-серверу о наличии установленного PHP. Установка PHP возможна двумя вариантами: как модуль Apache и как внешнее CGI-приложение. Ниже будут рассмотрены оба варианта установки.

Установка PHP в качестве модуля Apache

Установка PHP в качестве модуля немного повышает быстродействие, так как модуль PHP загружается один раз при запуске Web-сервера

> *Замечание*
>
> При установке PHP в качестве модуля настройки из php.ini читаются один раз при запуске Web-сервера. Поэтому при внесении изменений в php.ini необходимо перегрузить Apache для того, чтобы внесенные изменения вступили в силу.

Для установки PHP откройте файл главный настроечный файл Apache httpd.conf на редактирование и удалите символы комментариев со следующих строк, при необходимости изменив их:

```
AddType application/x-httpd-php phtml php
LoadModule php5_module c:/php/php5apache2.dll
```

> *Примечание*
>
> Вместо директории c:/php подставьте Вашу директорию с установленным PHP.

Установка PHP, как CGI-приложения

При установке PHP, как CGI-приложения интерпретатор PHP будет загружаться каждый раз при вызове PHP-сценария. В связи с этим, возможно, некоторое ухудшение быстродействия. Если PHP установлен, как CGI, то при внесении изменений в файл php.ini Apache перезагружать не следует, так как установки читаются каждый раз при выполнении PHP-сценария. Установка PHP как CGI немного ускоряет внесение изменений в конфигурацию PHP, так она не требует перезагрузки WEB-сервера.

Примечание

При установке PHP, как CGI перестанут работать некоторые заголовки, например, Вы не сможете организовать авторизацию пользователей средствами PHP. Авторизации можно будет реализовать только средствами самого Apache с помощью файлов .htaccess.

Для установки PHP откройте главный настроечный файл httpd.conf на редактирование, найдите в нем закомментированные строки подключения PHP и измените их следующим образом:

```
AddType application/x-httpd-php phtml php


<Directory "c:/php">
  Options ExecCGI
</Directory>


ScriptAlias "/php_dir/" "c:/php/"
Action application/x-httpd-php "/php_dir/php-
cgi.exe"
```

Примечание

Вместо директории c:/php подставьте Вашу директорию с установленным PHP.

Конфигурирование PHP (файл php.ini)

Так как на локальной машине вы, скорее всего, будете заняты тестированием Ваших Web-приложений, то необходимо должным образом настроить конфигурационный файл php.ini. Найдите директиву error_reporting и установите для неё следующее значение:

```
error_reporting = E_ALL & ~E_NOTICE
```

Это значение настроит PHP таким образом, что при работе PHP-скриптов будут отображаться все ошибки, а "замечания" будут игнорироваться. Так же обходимо проследить, чтобы директива display_errors, была включена:

```
display_errors = On
```

Если данная директива отключена (Off), то сообщения об ошибках не будут выводится в окно браузера и в случае возникновения в коде ошибки вы будете гадать перед девственно белым окном — что бы это означало. Так же необходимо проследить, чтобы директива variables_order имела следующее значение:

```
variables_order = "EGPCS"
```

Буквы здесь означают следующее:
E - переменными среды
G - переменными передаваемыми по методу GET (G)
P - переменными передаваемыми по методу POST (P)
C - Cookies
S - сессии
Отсутствие какой-либо из букв не позволит вам работать с соответствующими переменными.

Следующая директива, которая может потребовать настройки – это *register_globals*. Если данная директива включена

```
register_globals = On
```

то переменные передаваемые метором GET, POST, через cookies и сессии можно использовать в PHP-скрипте, обращаясь к ним просто как обычным переменным (`$someone`).

Если данная директива отключена

```
register_globals = Off
```

то к таким переменным можно будет обращаться только при помощи суперглобальных массивов ($_POST, $_GET и т.п.).

Директива *register_long_arrays* позволяет использовать суперглобальные массивы в старом формате ("длинном" — $HTTP_GET_VARS, $HTTP_POST_VARS и т.д.)

```
register_long_arrays = On
```

Теперь необходимо настроить индексный файл. Если в окне браузера набрать строку http://localhost/, а не http://localhost/index.html. Сервер всё равно предоставит браузеру index.html, так как этот файл является индексным и ищется в директории первую очередь, если не указан конкретный файл. Теперь необходимо настроить http.conf, таким образом, чтобы Web-сервер Apache так же реагировал на файлы index.php. Для этого найдте в http.conf директиву DirectoryIndex и исправьте её следующим образом:

```
DirectoryIndex index.html index.html.var index.php
```

После этого необходимо перегрузить сервер Apache, а в корневой директории виртуального хоста ("C:/www/") создать пробный файл PHP (index.php):

```
<?php
  phpinfo();
?>
```

В случае успешной настройки, обращение по адресу http://localhost/index.php отобразит фиолетовую таблицу с текущими настройками PHP, которая выдаётся функцией phpinfo().

Таким образом, у нас настроена связка Apache и PHP и можно переходить к настройке MySQL. Распакуйте дистрибутив MySQL во временную директорию и запустите установщик. Контролировать работу сервера

MySQL можно точно так же как и Apache, используя консоль управления сервисов Windows.

Установка расширений PHP

На последок вам возможно понадобится настроить некоторые расширения PHP, они настраиваются точно так же как и MySQL.

Так для того, чтобы подключить графическую библиотеку GDLib в php.ini необходимо раскомментировать строку:

```
extension=php_gd2.dll
```

Проверьте после этого наличие данной библиотеки в папке c:phpext. После внесения изменений в php.ini перезапустите сервер. Что бы быстро проверить: подключилась ли библиотека — выполните функцию phpinfo(). Если все в порядке, то в таблице, которая отображается функций phpinfo(), должен появится раздел "gd".

Некоторые расширения требуют дополнительных библиотек. Так для того чтобы воспользоваться расширением PHP "Mcrypt Encryption", позволяющем осуществлять симметричное шифрование необходимо, во первых, расскомментирования строку в php.ini

```
extension=php_mcrypt.dll
```

А во вторых скопировать в папку `C:/WINDOWS/` дополнительную библиотеку libmcrypt.dll

Список рекомендуемой литературы:

1. *Ветров С.* Операционная система MS Windows XP // М.: Солон-Р, 2001
2. *Курячий Г., Маслинский К.* Операционная система Linux. Курс лекций // М.: Интуит.ру, 2005
3. *Олифер В., Олифер Н.* Сетевые операционные системы — СПб: Питер, 2002
4. *Столлингс В.* Компьютерные сети, протоколы и технологии Интернета — СПб: BHV-СПб, 2005
5. *Таненбаум Э.С.* Современные операционные системы 2-е изд. — СПб: Питер, 2002
6. *Таненбаум Э.С.* Компьютерные сети. 4-е изд. — СПб.: Питер, 2003.