

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА РФ**  
**Федеральное государственное образовательное учреждение высшего**  
**профессионального образования**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**  
**УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ**

---

**Кафедра вычислительных машин, комплексов,**  
**систем и сетей**

**Н.И. РОМАНЧЕВА**

**БАЗОВЫЕ ИНТЕРНЕТ-ТЕХНОЛОГИИ**

**УЧЕБНОЕ ПОСОБИЕ**

**Москва- 2008**

Рецензенты: докт. техн. наук, доцент Егорова А.А.;  
зам. технического директора Дирекции по развитию  
ЗАО «Сирена-Трэвел» Белов С.Д.

Романчева Н.И.

Базовые Интернет-технологии. Учебное пособие - М.: МТУ ГА, 2008.-  
96 с.: Ил. 8 , список лит. 14 наим.

Данное учебное пособие содержит материал первой части дисциплины «Интернет-технологии в ГА», в которой рассматриваются базовые Интернет-технологии, в том числе технологии агрегирования адресов и взаимодействия с интерактивным конечным пользователем. В конце каждого раздела учебного пособия приводятся контрольные вопросы, в приложении - тестовые вопросы. Данное учебное пособие содержит материал, необходимый студентам для использования Internet в своей практической деятельности.

Учебное пособие издается в соответствии с учебным планом для студентов специальности 230101 дневного обучения.

Рассмотрено и одобрено 04.03.2008 г. на заседаниях кафедры ВМКСС и методического совета по специальности 230101 04.03.2008 г.

## СОДЕРЖАНИЕ

Введение	
Раздел 1. Технология агрегирования адресов CIDR	7
1.1 Проблемы адресации в IP-сетях	7
1.2 Методы перехода от IPv4 к IPv6	8
1.3 Особенности адресации IPv6	8
1.4 Форма записи IPv6	10
1.5 Типы адресов	11
1.6 Структура глобального агрегируемого уникального адреса	16
1.7 Формат заголовков IPv6	18
1.8 Изменения в DNS	20
Контрольные вопросы	21
Раздел 2. Базовые информационные технологии Интернет	22
2.1 Технологии взаимодействия с интерактивным конечным пользователем	22
2.2 TELNET как технология удаленного доступа к ресурсам сети	32
2.3 Технологии отложенного просмотра	36
Контрольные вопросы	44
Раздел 3. WEB-технологии	45
3.1 Основные компоненты Web- технологии	45
3.2 Использование модели «клиент-сервер» для WWW	47
3.3 Основные компоненты технология Active-X	49
3.4 Технология PHP	50
3.5 Технология CGI	57
3.6 Технология Java	60
3.7 Технология CSS	61
3.8 Технология поиска информации в Интернет	63

3.9 Информационные порталы . . . . .	74
Контрольные вопросы. . . . .	88
Список используемых источников . . . . .	90
Приложение. Контрольные тесты . . . . .	91

## ВВЕДЕНИЕ

Информационные технологии - это система приемов, способов, методов осуществления информационных процессов - получения, хранения, транспортировки (то есть передачи на расстояние), преобразования и представления информации.

Сегодня процесс проникновения технологий Интернет в различные сферы человеческой деятельности очевиден. В современном мире организация дискуссии для обмена информации в сообществе Интернет между учеными, разработчиками, пользователями, производителями оборудования и услуг, сетевыми администраторами не обходится без информационных и сетевых технологий.

В настоящее время идет естественный процесс эволюции Интернета вообще, и базовых технологий в частности.

Одной из задач IETF (в соответствии с RFC 4677) является содействие широкому распространению технологий и стандартов, разрабатываемых в Internet Research Task Force (IRTF). Группа технических советников IAB (Internet Architecture Board) постоянно осуществляет консультации руководства ISOC по техническим, архитектурным и процедурным вопросам, связанным с Интернет и его технологиями.

В основе сети Интернет лежит множество различных технологий, однако наиболее важной является возможность использования различных портов. Благодаря разным портам трафик электронной почты не оказывает влияние на данные, получаемые в процессе загрузки web-страницы или работы с удаленным сервером, хотя сами данные пересылаются по одному и тому же каналу связи.

Web- технологии, сегодня, позволяют создавать Интернет-проекты самого разного типа, сложности, а также целевой направленности. Они,

как и любые другие разработки, постоянно совершенствуются и развиваются, добавляются новые, заменяются устаревшие.

В первом разделе данного учебного пособия рассмотрены основные проблемы IP-адресации, методы перехода, в том числе с использованием технологии агрегирования адресов, форматы адреса IPv6.

Во втором разделе рассматриваются технологии взаимодействия с интерактивным конечным пользователем и технологии взаимодействия с интерактивным конечным пользователем.

Третий раздел посвящен Web-технологиям, приведена их классификация и основные компоненты. Большое внимание уделено технологиям поиска информации в Интернет, рассмотрены механизмы создания информационных порталов.

В конце каждого раздела учебного пособия приводятся контрольные вопросы, в приложении приведены тестовые вопросы.

Данное учебное пособие предназначено для студентов специальности 230101 и других специальностей для изучения вопросов, связанных с использованием базовых Интернет-технологий в своей практической деятельности.

## РАЗДЕЛ 1. ТЕХНОЛОГИЯ АГРЕГИРОВАНИЯ АДРЕСОВ

### 1.1 Проблемы адресации в IP-сетях

Адресация в сетях имеет большое значение. Учитывая ускоряющийся рост сети, ограниченность адресного пространства, использующего 32-х битный IP-адрес, является слабым местом протокола IPv4. Другой проблемой стала недостаточная масштабируемость маршрутизации – процедуры, составляющей суть IP-сетей. Обработка таблицы маршрутизации, содержащей информацию о нескольких десятках тысяч номеров сетей, вызывает перегрузку маршрутизатора.

В результате сообщество Интернет выбрало в качестве новых целей модернизации протокола IP: создание масштабируемой схемы адресации; повышение пропускной способности сети за счет сокращения работ, выполняемых маршрутизаторами; обеспечение защиты данных, передаваемых по сети; предоставление гарантий качества транспортных услуг.

Для решения этой проблемы разработан протокол IP нового поколения Next Generation Internet Protocol, IPng – IPv6, в котором для IP-адреса используется 128 бит.

Стандарт IPv6 является прямым преемником четвертой версии Интернет-протокола. Разрыв в последовательности номеров объясняется тем, что согласно рекомендациям RFC 1819 версия 5 была зарезервирована для экспериментального потокового протокола (streaming protocol). С начала 1990-х гг., когда стало ясно, что из-за расширения Интернет в обозримом будущем появятся различные проблемы, комитет Internet Engineering Task Force (IETF) начал работу над протоколом-преемником (RFC 2460). Наряду с ликвидацией ограничений по объему адресного пространства, были предусмотрены дальнейшие усовершенствования и оптимизация IP-протокола.

## 1.2 Методы перехода от IPv4 к IPv6

Разработчики IPv6 стремились предусмотреть механизмы плавного перехода с IPv4. Под *переходом* (transition) IETF понимает любые методы перевода конечных устройств, маршрутизаторов и сетей существующей инфраструктуры, базирующейся на IPv4, на новую версию IPv6. При этом первоочередная цель - обеспечить дальнейшее взаимодействие IPv6 совместимых устройств с традиционным IPv4-оборудованием.

В настоящее время из многообразия подходящих вариантов перехода можно выделить:

- методы на основе двойного набора протоколов (dual stack): сетевые компоненты снабжены двумя стеками протоколов, т. е. поддерживают как IPv4, так и IPv6;

- методы туннелирования, предусматривающие упаковку данных IPv6 в заголовки IPv4 (и наоборот): становится возможной передача трафика по сетям, которые сами по себе тот или иной «туннелируемый» протокол не поддерживают;

- трансляция IPv4 в IPv6 и обратно, обеспечивающая взаимодействие оборудования IPv6 с системами на основе IPv4.

## 1.3 Особенности адресации IPv6

Адрес IPv6 состоит из 128 бит или 16 байт. Это дает возможность пронумеровать огромное количество узлов. Такое значительное увеличение длины адреса было сделано только и даже не столько для снятия проблемы дефицита адресов, сколько для дальнейшего повышения эффективности работы всего стека TCP/IP.

Главной целью изменения системы адресации было не механическое увеличение разрядности адреса, а изменение его функциональности за счёт введения новых полей.

Вместо прежних двух уровней иерархии адреса (номер сети и номер узла) IPv6 предлагается использовать четыре уровня, включая трехуровневую идентификацию сетей, и один уровень для идентификации узлов сети. За

счет увеличения числа уровней иерархии в адресе новый протокол эффективно поддерживает технологию агрегирования адресов CIDR (Classless InterDomain Rounding). Бесклассовая адресация CIDR - метод IP-адресации, позволяющий гибко управлять пространством IP-адресов, не используя жёсткие рамки классовой адресации. Суть этой технологии состоит в том, что каждому провайдеру выделяется непрерывный диапазон в пространстве IP-адресов (напомним, что в IPv4 существует 5 классов адресов (A, B, C, D, E)). Бесклассовая адресация основывается на переменной длине маски подсети VLSM (Variable Length Subnet Mask), в то время, как в классовой адресации длина маски строго фиксирована 0, 1, 2 или 3 установленными байтами. Благодаря этому, а также усовершенствованной системе групповой адресации и введению адресов типа *anycast*, новая версия IP позволяет уменьшить затраты на маршрутизацию.

Версия 6 вносит некоторые изменения в терминологию версии 4 и вводит новые термины:

- *Пакетом* (packet) называется заголовок IPv6 плюс полезные данные;
- *Узел* (node) - любая система, реализующая протокол IPv6;
- *Маршрутизатор* (router) - узел, пересылающий не адресованные ему пакеты IPv6;
- *Связь* (link) - носитель, по которому взаимодействуют узлы на уровне связи данных;
- *Соседи* (neighbor) - узлы, подключенные к одной связи.

Одно из новшеств в IPv6 - это возможность переноса трафика для многих других протоколов. Следовательно, полезные данные не обязательно будут PDU из набора протоколов TCP/IP. *Пакетами* называются любые элементы данных протокола (PDU) от уровня связи данных до уровня приложений [11].

Базовый набор протоколов IPv6 был принят IETF в сентябре 1995 года. В августе 1998 года приняты пересмотренные версии группы стандартов, определяющих как общую архитектуру IPv6 (RFC 2460 Internet Protocol, Version 6 (IPv6) Specification), так и его отдельные аспекты - например, систему адресации (RFC 2373 "IP Version 6 Addressing Architecture").

#### 1.4 Форма записи IPv6

Адреса IPv6 имеют длину 16 октетов (128 бит). Для записи адресов используется компактная нотация. Адреса представлены как 8 шестнадцатеричных чисел, разделенных двоеточиями. Каждое шестнадцатеричное число представляет 16 бит. Например:

*32BD:0000:10A12:0006:0001:DDE1:8006:2334*

Ведущие нули в шестнадцатеричных полях могут быть опущены (например, 0 вместо 0000 и 6 вместо 0006 и т.д.):

*32BD:0:10A12:6:1:DDE1:8006:2334*

Формат может быть еще более сжат при замене последовательности смежных нулевых полей на "::". Например, адрес

*32BD:0:0:0:1:DDE1:8006:2334*

может быть заменен на адрес

*32BD::1:DDE1:8006:2334*

Сокращение «::» можно использовать только один раз.

Адреса версии 4 протокола IP часто вкладываются в последние четыре октета версии 6. Они могут быть записаны с использованием смешанного формата адреса (сочетающего как нотацию с точками, так и нотацию с двоеточиями), например:

*0:0:0:0:0:FFFF:226.3.4.101*

128-разрядное пространство IP-адреса версии 6 обеспечивает место для множества различных типов адресов, включая:

- иерархические глобальные одноадресные рассылки на основе адресов провайдеров;

- иерархические глобальные одноадресные рассылки по географическому признаку;
- личные адреса сайтов для использования только в пределах организации;
- локальные и глобальные многоадресные рассылки.

Версия 6 не использует широковещательные рассылки, но для функций управления (например, разрешения адресов или загрузки) использует многоадресные рассылки. Это связано с тем, что сообщения широковещательных рассылок прерывают работу всех устройств связи, хотя в большинстве случаев они предназначаются лишь для небольшого количества устройств. Кроме того, ограничение управляющих сообщений только многоадресными рассылками предотвращает взаимовлияние устройств версии 6 и версии 4, совместно использующих одну и ту же связь.

### 1.5 Типы адресов

Протоколом IPv6 определено 3 основных типа адресов:

unicast;                    multicast;                    anycast;

Тип адреса задается значением нескольких старших бит адреса, которые названы *префиксом формата*.

Адрес типа *unicast* определяет уникальный идентификатор отдельного интерфейса конечного узла или маршрутизатора. Назначение этого типа адреса совпадает с назначением уникальных адресов в версии IPv4 - с их помощью пакеты доставляются определенному интерфейсу узла назначения. В версии IPv6, в отличие от версии IPv4, как говорилось ранее, отсутствует понятие класса сети и связанное с ним фиксированное разбиение адреса на номер сети номер узла по границам байт.

Адреса типа unicast делятся на несколько подтипов для отражения специфики некоторых ситуаций, часто встречающихся в современной мультисервисной сети.

Адрес типа *multicast* - групповой адрес, аналогичный по назначению групповому адресу. Он имеет префикс формата 1111 1111 и идентифицирует группу интерфейсов, относящихся, как правило, к разным узлам. Пакет с таким адресом доставляется всем интерфейсам с этим адресом. Адреса типа *multicast* используются в IPv6 и для замены широковещательных адресов. Для этого вводится особый адрес группы, объединяющей все интерфейсы сети.

В версии сии IPv6 групповой адрес имеет признак обзора (*scope*), отсутствовавший в групповом адресе версии IPv4. Этот признак позволяет гибко задавать область действия группового адреса. Она может представлять собой, например, только одну подсеть, либо все подсети данного предприятия, либо весь Интернет. Это упрощает работу маршрутизаторов, которым необходимо выявить все узлы, относящиеся к какой-либо группе. Еще один признак задает тип группы - постоянная или временная.

Адрес типа типа *anycast* - это новый тип адреса, который так же, как и *multicast*, определяет группу интерфейсов. Но пакет с таким адресом доставляется из интерфейсов группы, как правило, "ближайшему" в соответствии с метрикой, используемой протоколами маршрутизации. Синтаксически *anycast* адрес ничем не отличается от адреса типа *unicast*, он назначается из того же диапазона адресов, что и *unicast*-адреса. Следует подчеркнуть, что адрес типа *anycast* может назначен только интерфейсам маршрутизатора. Интерфейсы маршрутизаторов, входящие в одну *anycast*-группу, имеют индивидуальные *unicast*-адреса и, кроме того, общий *anycast*-адрес.

Адреса такого типа ориентированы на применение маршрутизации от источника (*Source Routing*), когда маршрут прохождения пакету определяется узлом-отправителем путем указания IP-адресов всех промежуточных маршрутизаторов.

В шестой версии протокола, как и в четвертой версии протокола

IP имеются адреса, предназначенные для локального использования, то есть в сетях, не входящих Интернет.

Напомним, что некоторые адреса IPv4 являются выделенными и трактуются по-особому:

все нули		данный узел
номер сети	все нули	
все нули	номер узла	данная IP-сеть
все единицы		
номер сети	все единицы	узел в данной (локальной) IP-сети
127	что-нибудь	
		все узлы в данной (локальной) IP-сети
		все узлы в указанной IP-сети

Как видно в выделенных IP-адрес«петля» либо данному узлу, либо данной IP-сети, а IP-адреса, состоящие из всех 1, используются при широковещательных передачах (multicast). Для ссылок на всю IP-сеть в целом используются IP-адрес с нулевым номером узла. Особый смысл имеет IP-адрес, первый октет которого равен 127. Он используется для тестирования программ и взаимодействия процессов в пределах одной машины. Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется как бы «петля». Данные не передаются по сети, а возвращаются модулям верхнего уровня, как только что принятые. Поэтому в IP-сети запрещается присваивать машинам адреса, начинающиеся с 127.

При разработке структуры IP-адресов предполагалось, что они будут использоваться по разному назначению. Адреса класса А предназначены для использования в больших сетях общего пользования. Адреса класса В предназначены для использования в сетях среднего размера (сети больших компаний, научно-исследовательских институтов, университетов). Адреса класса С предназначены для использования в сетях с небольшим числом компьютеров (сети небольших компаний и фирм).

Для внутрикорпоративного использования Оргкомитет Интернет выделил, так называемые, Intranet IP-numbers:

- одну сеть класса А с номерами 10.\*.\*.\* ;

- 16 сетей класса В с номерами 172.[16...31].\*.\* ;
- 256 сетей класса С с номерами 192.186.\*.\* .

В отличие от четвертой версии, в шестой эти адреса имеют специальный формат. Адреса для локального использования представлены в IPv6 двумя разновидностями. Во-первых, это адреса для сетей, не разделенных на подсети (не использующие маршрутизацию). Они называются Link-Local (локальных связей). Во вторых, это адреса локального использования для сетей, разделенных на подсети. Такие адреса называются Site-Local (локальных сайтов).

Адреса Link-Local имеют формат:

1111111010 (10 бит)	00...00	Уникальный адрес технологии связи
---------------------	---------	-----------------------------------

Как видно из приведенного формата, адрес Link-Local имеет 10-битный префикс и содержит только 64-разрядное поле идентификатора интерфейса. Остальные разряды, кроме префикса, должны быть нулевыми, поскольку потребность в номерах подсети отсутствует. Адреса локальных связей весьма полезны при инициализации.

Если сайт имеет маршрутизаторы, но не связан с провайдером, можно автоматически генерировать внутренние адреса типа Site-Local. Данный тип адреса имеет формат:

1111111011(10 бит)	00...00	Идентификатор сети	Уникальный адрес технологии связи
--------------------	---------	--------------------	-----------------------------------

По сравнению с адресом Link-Local адреса Site-Local содержат дополнительное двухбайтовое поле номера подсети.

В данном случае префикс предоставляют маршрутизаторы (включая идентификатор подсети). В качестве уникального адреса технологии связи можно использовать, например, MAC-адрес локальной сети.

От этого формата легко перейти к соединению с провайдером – маршрутизатор просто конфигурируется с новым префиксом, который содержит идентификаторы регистратора адреса, провайдера и подписчика вместе с номером подсети. Маршрутизатор предоставляет новый префикс, а хосты начинают им пользоваться. Назначенная в сайте часть адреса не меняется.

Многоадресные рассылки в версии 6 имеют более четкое и гибкое определение, чем в версии 4. Введено множество типов таких рассылок. Они немного различаются в зависимости от своих свойств: постоянный адрес (permanent), кратковременный (transient), локальный (local) или глобальный (global).

Многоадресные рассылки имеют следующий формат:

8 бит	4 бит	4 бит	112 бит
11111111	000T	Вложенность	Идентификатор группы

Где T=0 для общеизвестного постоянного адреса многоадресной рассылки, и T=1 для кратковременного адреса многоадресной рассылки.

Коды вложенности указывают, находится ли область действия в пределах того же узла, локальной связи, локального сайта, данной организации или это глобальная структура. Область действия внутри узла охватывает и случай, когда клиент посылает сообщение многоадресной рассылки на сервер, находящийся в том же самом хосте. В [10] полностью определены коды вложенности. В данном пособии приведем лишь ряд кодов. Например, для локального узла код вложенности равен 1, для локального сайта – 5, внутри организации – 8, глобально – E, F – зарезервировано.

В IPv6 существуют «неспецифицированные адреса» (unspecified address) – адреса со всеми нулями:

0:0:0:0:0:0:0

Они иногда используются как адрес источника во время инициализации, когда система еще не знает собственного адреса.

Кольцевые адреса (loopback) в версии IPv6 определены следующим образом:

0:0:0:0:0:0:0:1

В смешанном окружении адресов версий 4 и 6 IP-адреса систем версии 4, которые не поддерживают версию 6, отображаются в адреса версии 6 следующим образом:

0:0:0:0:FFFF:a.b.c.d ,

где a.b.c.d - исходный IP-адрес.

Таким образом, эти адреса легко отображаются между их представлениями в версиях 4 и 6.

Работа по детализации подтипов адресов IPv6 еще далека от завершения. Сегодня определено назначение лишь 15% адресного пространства IPv6. Оставшаяся часть адресов не определена.

### 1.6 Структура глобального агрегируемого уникального адреса

Основным подтипом адреса типа unicast является глобальный агрегируемый уникальный адрес, который предназначен для идентификации узлов в Интернет. Такие адреса могут агрегироваться для упрощения маршрутизации. Глобальные агрегируемые уникальные адреса IPv6 имеют более сложную структуру, по сравнению с четвертой версией, и включает шесть полей:

3	13	8	24	16	64
FR	TLA	-	NLA	SLA	Interface ID

1) Поле префикаса (FR) для этого типа адресов состоит из трех бит и имеет значение 001.

Следующие три поля TLA, NLA, SLA рассматриваются как префиксы трех уровней – верхнего уровня агрегирования адресов, следующего уровня агрегирования и местного уровня агрегирования.

2) Префикс верхнего уровня TLA (Top-Level Aggregation) предназначен для идентификации провайдеров самого верхнего уровня. Конкретное значение этого префикса представляет собой общую часть адресов, которыми располагает данный провайдер. Сравнительно небольшое количество разрядов, отведенных под поле (т.е. 13), выбрано специально для ограничения размера таблиц маршрутизации в магистральных маршрутизаторах самого верхнего уровня Интернет. Это поле позволяет перенумеровать 8196 сетей провайдеров верхнего уровня. Следовательно и число записей, описывающих маршруты между этими сетями, также будет ограничено значением 8196, что ускорит работу магистральных маршрутизаторов.

3) Следующие 8 разрядов зарезервированы для расширения в будущем поля TLA, если 8196 номеров сетей окажется недостаточно на верхнем уровне.

4) Префикс следующего уровня NLA (Next-Level Aggregation) предназначен для размещения номеров сетей средних и мелких провайдеров. Значительный размер поля NLA позволяет путем агрегирования адресов создавать их многоуровневую иерархию, отражающую многоуровневую иерархию провайдеров.

5) Префикс местного уровня SLA (Site-Level Aggregation) предназначен для адресации подсетей отдельного абонента, например, подсетей одной корпоративной сети. Предполагается, что провайдер назначает некоторому предприятию номер его сети, состоящий из фиксированного значения полей TLA и NLA, которые в совокупности являются аналогом номера сети версии IPv4. Остальная часть адреса - поля SLA и идентификатор интерфейса - поступают в распоряжение администратора корпоративной сети, то есть он полностью управляет

процессом его формирования и не должен согласовывать этот процесс «провайдером. Причем поле идентификатора интерфейса имеет вполне определенное назначение - оно должно хранить физический адрес узла.

На этом уровне также возможно агрегировать адреса небольших подсетей в более крупные подсети, и размер поля в 16 бит обеспечивает достаточную свободу и гибкость построения внутрикорпоративной иерархии адресов.

б) Идентификатор интерфейса (Interface ID) является аналогом номера узла IPv4. Идентификаторы интерфейсов используются в адресах типа unicast для однозначного определения интерфейсов в пределах какой-либо подсети. Отличием версии IPv6 является то, что в общем случае идентификатор интерфейса просто совпадает с его локальным адресом, а не представляет собой произвольно назначенный администратором номер узла. Interface ID имеет длину 64 бита. Это позволяет поместить туда MAC-адрес (48 бит), или адрес X.25 (до 60 бит), или адрес конечного узла АТМ (48 бит), или номер виртуального соединения АТМ (до 28 бит), а также дает возможность использовать локальные адреса технологий, которые могут появиться в будущем.

Такой подход делает ненужным использование протокола ARP.

### **1.7 Формат заголовков IPv6**

В протоколе IPv6 существует несколько заголовков. Основной заголовок очень прост и имеет немного полей:

- *Version* – версия, равна 6 для IP следующего поколения;
- *Priority*– приоритет - определяет последовательность отбрасывания во время перегрузки
- *Payload length* - длина полезной нагрузки (16 бит). Если длина меньше или равна 64 Кбит, это поле сообщает о длине части пакета, следующей за начальным заголовком IPv6. Если длина больше 64 Кбит, длина полезной нагрузки указывается как нулевая, а фактическая длина

будет сообщена *Jumbo Payload* (гигантская полезная нагрузка) в следующем далее заголовке.

- *Hop limit* - предел для попаданий, уменьшается на единицу в каждом маршрутизаторе. Пакет будет отброшен, когда значение достигнет нуля.

- *Next header* - следующий заголовок, идентифицирует тип следующего далее заголовка протокола (например, 6 для заголовка TCP),

- *Flow label* - метка потока, указывает на трафик со специальными свойствами (например, видео в реальном времени).

Использование дополнительных заголовков (*extension header*) в IPv6-это прогрессивная идея, позволяющая последовательно добавлять новые функциональные возможности.

Напомним, что в заголовке IP версии 4 поле протокола служит для идентификации типа заголовка (например, TCP или UDP), следующего за заголовком IP. Версия 6 использует более общее поле *Next Header*. Если следующий заголовок определен как TCP или UDP, значением поля будет 6 или 17.

Между заголовком IPv6 и заголовком верхнего уровня можно вставить несколько дополнительных заголовков для необязательных вариантов, подобных маршрутизации от источника или поддержке безопасности. Фрагментация также может быть перенесена в дополнительные заголовки.

Как показано на рисунке 1.1, каждый дополнительный заголовок имеет поле *Next Header*, что позволяет связать все эти заголовки в цепочку. Протокол следующего уровня идентифицируется включенным в общую последовательность дополнительным заголовком.

Такая схема обеспечивает большую гибкость. По мере необходимости могут определяться новые возможности, поскольку нет ограничений на общую длину. Заключительный дополнительный

заголовок может ссылаться на заголовок полностью независимого протокола, например ISO или DECnet.

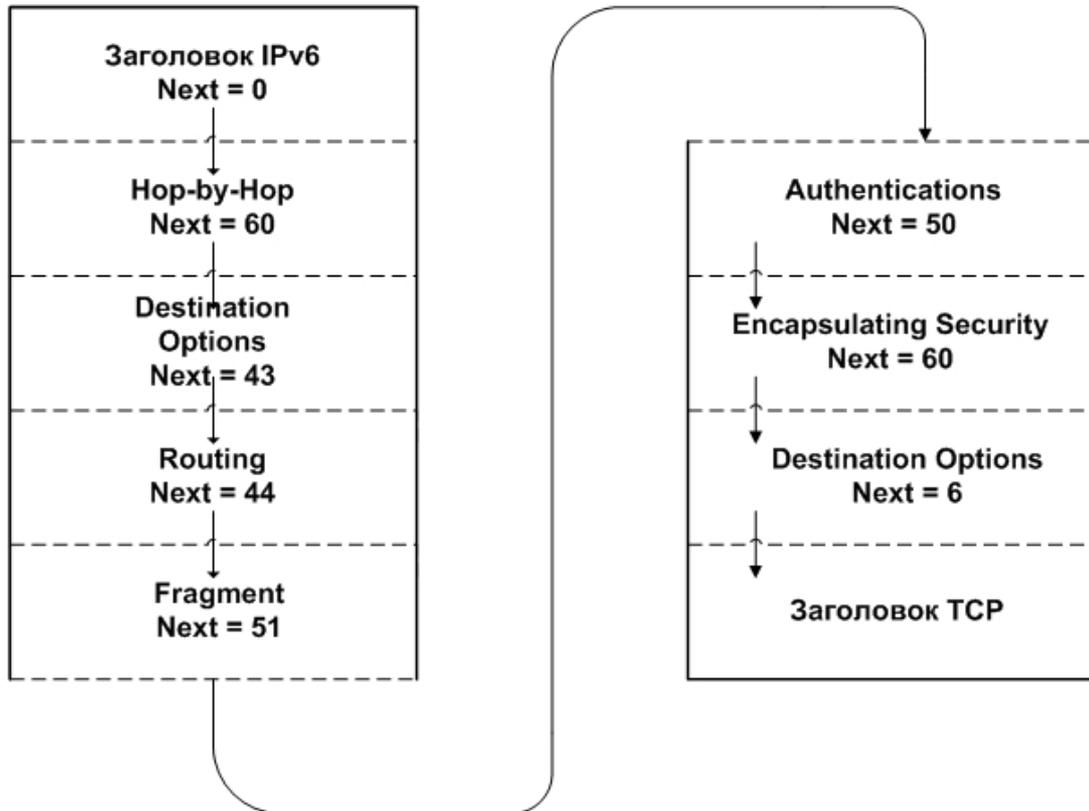


Рисунок 1.1 – Порядок цепочки дополнительных заголовков

Показанный на рисунке 1.1 порядок отражает рекомендации по размещению заголовков. Как видно из рисунка, могут присутствовать два заголовка Destination Options. Первый из них стоит перед Routing и применяется к каждому из попаданий, указанных в заголовке Routing. Второй присутствует как последний заголовок и применяется только в точке назначения.

Определенные на настоящий момент заголовки представлены в [8]. Некоторые из них содержат информацию, которую следует обрабатывать на каждом узле по пути следования пакета, в то время как другие заголовки обрабатываются только в точке назначения.

### 1.8 Изменения в DNS

Новый тип записи о ресурсе, *RRRR*, отображает имена доменов в адреса IP версии 6. Пример такой записи:

*SERGEY IN RRRR 5422:0:5:123:3:40:67:79BD*

Для преобразования адресов в имена для IPv6 потребуется добавить новые домены. Обратный поиск доменов включается в дерево доменов от узла *IP6.INT*.

Адреса IP версии 4 рассматриваются в обратном порядке, чтобы получить свои метки в домене *in-addr.arpa*. Адреса версии 6 также просматриваются наоборот и переписываются как ряд шестнадцатеричных цифр, разделенных точками. Например, обратная запись приведенного элемента появится в дереве домена как:

D.B.9.7.7.6.0.0.0.4.0.0.3.0.0.0.3.2.1.0.5.0.0.0.0.0.0.2.2.4.5.IP6.INI

**Контрольные вопросы**

- 1) Перечислите основные цели модернизации протокола IP.
- 2) Поясните суть технологии CIDR.
- 3) Перечислите основные типы адресов IPv6.
- 4) Назовите основные характеристики IPv6.
- 5) Поясните взаимодействие сетей IPv6 через промежуточные сети IPv4.
- 6) Поясните назначение «неспецифицированных адресов».
- 7) Поясните общие принципы выделения адресов в адресном пространстве IPv6 региональным организациям.
- 8) Перечислите поля основного заголовка IPv6.
- 9) Поясните цель использования дополнительных заголовков в IPv6.
- 10) Назовите адресные префиксы IPv6.
- 11) Какие изменения произошли в DNS?

## РАЗДЕЛ 2. БАЗОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ИНТЕРНЕТ

### 2.1 Технологии взаимодействия с интерактивным конечным пользователем

При работе в многосистемном сетевом окружении, каким является сеть Интернет, возникают проблемы, связанные: с различными правилами именования файлов и перемещения по каталогам файловой системы; ограничениями на доступ к файлам; с различными способами представления данных в файлах. Для решения этих проблем был создан протокол пересылки файлов FTP (File Transfer Protocol). На эффективность операций пересылки файлов влияют следующие факторы:

- файловая система хоста и производительность его дисков;
- объем обработки по переформатированию данных;
- используемая служба TSP.

В настоящее время существует ряд протоколов данного типа, из которых внимания заслуживают три – вышеупомянутый протокол FTP, SFTP (Simple File Transfer Protocol) и TFTP (Trivial File Transfer Protocol).

#### 2.1.1 Протокол FTP

Протокол FTP создан для взаимодействия с интерактивным конечным пользователем или прикладной программой. Протокол FTP - это совокупность соглашений, выполняемых при пересылке файлов в Интернет. Под протоколом FTP также понимаются средства доступа к удаленному компьютеру, позволяющие просматривать его каталоги и файлы, переходить из одного каталога в другой, копировать, удалять и обновлять файлы. Все эти функции являются частью стандартного стека протоколов TSP/IP [1]

С протоколом FTP связаны следующие понятия:

- команды и их параметры, передаваемые по управляющему соединению;

- числовые коды ответа сервера;
- формат пересылаемых данных.

В FTP также как и SMTP и POP3 команды, и ответы на них передаются в формате строк NVT ASCII. При обмене командами используется принцип последовательного выполнения. Однако в отличие от «почтовых» протоколов FTP открывает два TCP-соединения - одно для команд, а другое - для данных.

#### **а) Управление данными**

В FTP два TCP-соединения определяются как управляющее соединение и соединение данных. *Управляющее соединение* – типичное соединение клиент-сервер. Сервер FTP обеспечивает пассивное открытие на официальном порту и ждёт запроса на установление соединения от клиента. Клиент FTP, в свою очередь, входит в контакт с FTP-сервером на официальном порту протокола и устанавливает с ним TCP-соединение. Управляющее соединение остаётся активным на протяжении всего FTP-сеанса. Через управляющее соединение клиент и сервер обмениваются строками команд NVT ASCII и кодами ответа. FTP создаёт отдельное *соединение данных* для каждой операции по передаче файла (а также в некоторых других случаях). Основанные на протоколе FTP программы используют соединение данных для трех основных целей:

- чтобы послать список файлов или каталогов от сервера клиенту;
- чтобы послать файл от клиента серверу;
- чтобы послать файл от сервера клиенту.

В настоящее время в Интернет существует три разновидности FTP-серверов: Internet-style (доступ ко всем файлам сервера); Listserver (ограниченный доступ); FTPmail (доступ через электронную почту).

Серверы FTPmail наиболее интересны для тех пользователей, у которых доступ к Интернет весьма ограничен.

Рисунок 2.1 иллюстрирует типичную конфигурацию для операций по передаче файлов. Основа модели - интерпретаторы протокола (PI) и процессы передачи данных (DTP). Как видим, на стороне клиента и сервера имеются собственные PI и DTP. Процессы передачи данных устанавливают и управляют соединением данных. Интерпретаторы протокола интерпретируют FTP-команды и общаются через управляющее соединение. Интерфейс пользователя ограждает пользователя от непосредственного общения с командами и ответами FTP.

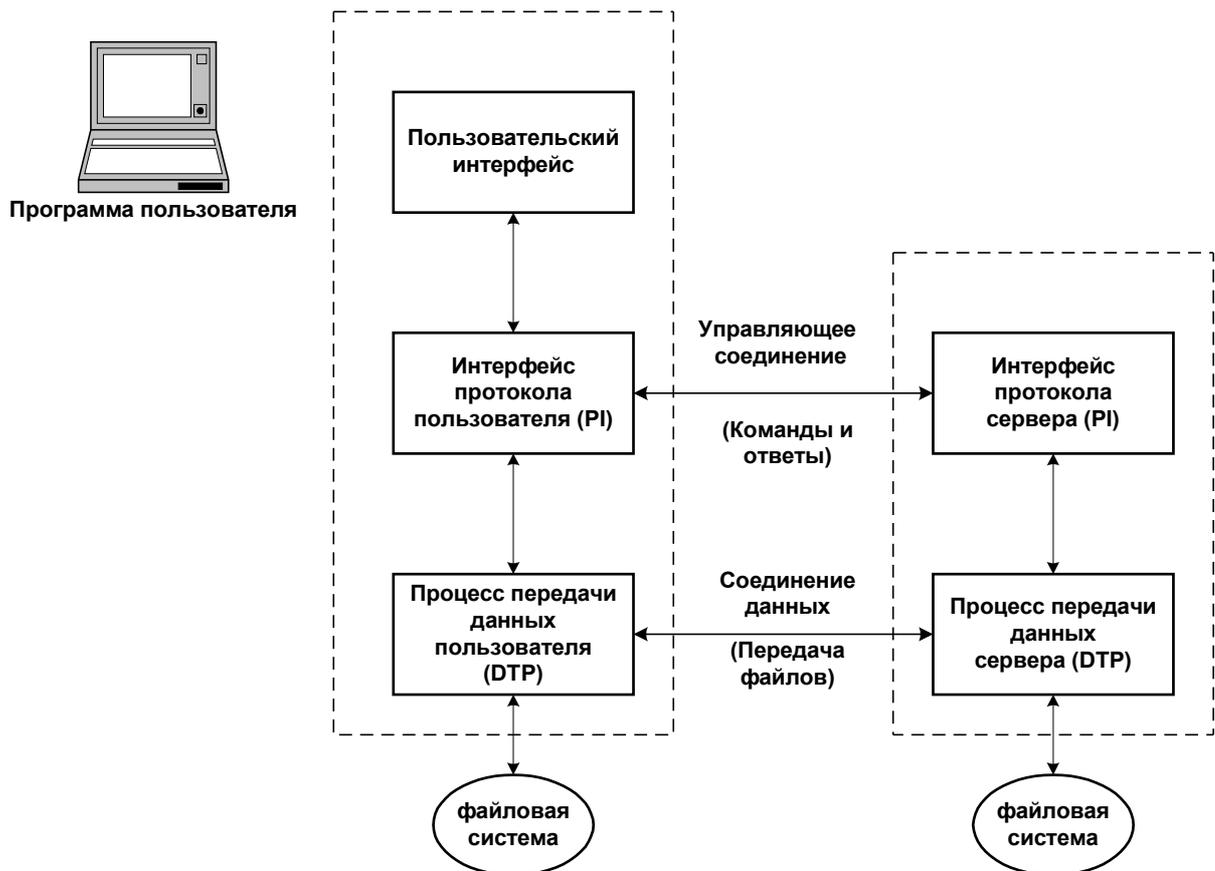


Рисунок 2.1- Модель FTP

С самого начала протокол FTP разрабатывался для работы с различными компьютерами, использующими различные операционные системы, структуры файлов и наборы символов. В результате FTP требует, чтобы пользователи выбрали соответствующие опции для операций по передаче файла.

### б) Опции FTP

Опции FTP можно разбить на 4 категории: тип файла, формат файла, структура файла, способы передачи.

1. Тип файла - FTP может управлять четырьмя различными *типами файлов*:

- локальным - предназначен для передачи файла между хостами, которые используют различные размеры байта;

- файлами изображений (или двоичными) - передаются как непрерывный поток данных;

- EBCDIC (Extended Binary Coded Decimal Interchange Code) - используются в некоторых системах, типа универсальных ЭВМ

- фирмы IBM (мэйнфреймы) и мини-компьютерах;

- ASCII - принимаемый по умолчанию тип для передачи файлов по FTP.

Основная проблема с передачей файла ASCII - маркеры конца строки (end, off-line и т.д.).

2. Формат файла - в FTP определяет три типа управления форматом:

- Nonprint - означает, что файл не содержит никакой информации о вертикальном формате типа вертикальной прокрутки страницы;

- Telnet - использует управление вертикальным форматом для принтеров;

- FORTRAN - использует специальные, вложенные в текст символы управления.

Сегодня большинство реализации FTP (особенно в UNIX-системах) ограничиваются управлением форматом Nonprint.

3. Режим передачи FTP - FTP определяет три режима передачи через TCP-соединение:

- блочный - файл передается как последовательность блоков, где каждый блок включает один или больше байтов заголовка;

- со сжатием - последовательные появления одного и того же байта кодируются специальным символом с указанием количества одинаковых символов в последовательности;

- потоковый - файл передается как непрерывный поток байтов.

3. Структура файла - если тип структуры FTP — *структура с записями*, FTP использует специальную двухбайтовую последовательность символов, чтобы отметить конец записи и конец файла. Если FTP использует *файловую структуру*, конец файла отмечается закрытием TCP-соединения, т.е. после того как передан последний байт файла, FTP закрывает соединение данных. Более распространена файловая структура, которая используется по умолчанию. Перейти на структуру записей можно стандартной командой *STRU R*, пересылаемой по управляющему соединению.

### **в) Типичный сеанс FTP**

Типичный сеанс FTP состоит из четырех стадий:

- программа клиент FTP соединяется с FTP сервером;
- пользователь регистрируется на FTP сервере;
- клиент и сервер обмениваются командами и ответными сообщениями подобно обмену между клиентом и сервером SMTP и POP3;
- FTP клиент закрывает соединение с сервером.

Ниже приведен типичный пример обмена FTP-командами клиента и сервера:

```

220-Local time is now 08:50. Server port: 21.
220-Only anonymous FTP is allowed here
220 You will be disconnected after 15 minutes of inactivity.
USER anonymous
230 Anonymous user logged in
PWD
257 "/" is your current location
REST 0
350 Restarting at 0
PASV
227 Entering Passive Mode (10,0,4,1,254,5)
LIST
150 Accepted data connection
226-Options: -l
226 1 matches total
QUIT
221 Good-bye

```

Клиент передает запросы через управляющее соединение, а не непосредственно через соединение данных. Такой принцип

функционирования выдвигает на первый план важное различие между FTP-клиентом и другими клиентами. В действительности, клиент и сервер обмениваются сообщениями через сокет, соединенный с сервером через официальный порт протокола. Следует обратить внимание на то, что передача данных FTP не происходит через официальный порт. Она происходит через порт, который выбирает сетевой компьютер клиента. Другими словами, для соединения данных FTP-клиент действует подобно серверу:

- клиент создает сокет;
- связывает его с местным адресом;
- сообщает серверу, какой адрес использовать, чтобы войти в контакт;
- ожидает входящее соединение.

Однако различие между FTP-клиентом и настоящим сервером в том, что первый принимает соединение только от FTP-сервера на другом конце управляющего соединения, а сокет сервера обслуживает запросы на установление соединения, пришедшие от любого удаленного компьютера. FTP-клиент хранит адрес FTP-сервера в сокете, созданном для соединения данных.

#### **г) Общие команды FTP**

За последние годы набор команд FTP, передаваемых по управляющему соединению существенно увеличился, однако хостам необязательно реализовывать все специфированные команды. Команды FTP можно разделить на три категории:

- *команды контроля доступа* - передают серверу информацию, идентифицирующую пользователя или сообщают серверу каталоги, к которым программа-клиент желает получить доступ (например, команды USER, PASS, REIN);

- *команды передачи параметров* - регистрируют тип, формат файла, структуру файла и режим передачи (например, PASV, PORT, STRU, TYPE, MODE);

- *команды обслуживания (сервиса)* - определяют операции по передаче файлов (например, LIST).

#### **д) Коды ответа сервера FTP**

Код ответа содержит три цифры. Каждая цифра в коде ответа имеет специальное значение. Смысл первой цифры (диапазон 1-5):

1 – предварительно положительный ответ, сервер начал требуемую операцию;

2 –положительный ответ завершения, сервер успешно закончил требуемое действие;

3 – промежуточный положительный ответ, сервер принял команду, но требуется дополнительная информация;

4 – временный отрицательный ответ завершения, сервер не принял команду, и требуемое действие не выполнялось;

5 – постоянный отрицательный ответ завершения, сервер не принял команду, и требуемое действие не выполнялось.

Вторая цифра в кодах ответа FTP идентифицирует сообщение более подробно (диапазон 0-5):

0 – *Синтаксис*: эти ответы относятся к ошибкам синтаксиса и синтаксически правильным командам, которые не попадают ни в одну из групп данной классификации;

1 – *Информация*: ответы на запрос о дополнительной информации, например информации о состоянии или помощи;

2 – *Соединение*: ответы относятся к управляющему соединению или соединению данных;

3 – *Авторизация и учет использования ресурсов*: - ответы на регистрацию в системе и процедуру учета;

4 - *Не определен*.

5 - *Файловая система*: ответы информируют о состоянии файловой системы сервера в ответ на запрос файловой системы.

Например, в ответ на команду клиента USER сервер отвечает кодом 331, т.е. первая цифра 3 означает, что сервер принял команду и требуется дополнительная информация. Вторая цифра 3 означает, что эта информация связана с регистрацией. Как правило, после команды USER следует команда PASS.

Спецификация FTP (RFC 959) предполагает явно знание транспортного протокола TCP и протокола TELNET. FTP использует протокол TELNET на управляющем соединении для управления действиями по передаче файла. Хотя FTP зависит от протокола TELNET, он использует лишь небольшое подмножество его команд.

Некоторые серверы не могут контролировать управляющее соединение и соединение данных одновременно. Спецификация FTP (RFC 959) определяет следующую процедуру для привлечения внимания FTP-сервера во время выполнения операций передачи файла:

- 1) Система пользователя вставляет сигнал TELNET IP (*прерывание процесса*) в поток TELNET (через управляющее соединение).
- 2) Система пользователя посылает сигнал *Synch* в сегменте, который содержит только байт команды DATA MARK TELNET.
- 3) Система пользователя вставляет команду FTP (например, ABOR) в поток TELNET (через управляющее соединение).
- 4) Интерпретатор протокола сервера (PI) после получения байта команды IP просматривает поток TELNET, чтобы получить *единственную* команду FTP.

Для получения файлов из Интернет можно использовать универсальные программы-браузеры, но в некоторых случаях удобнее воспользоваться специальной программой для передачи файлов, например, CuteFTP [5].

### **2.1.2. Простой протокол передачи файлов TFTP**

TFTP (Trivial File Transfer Protocol) — простой протокол передачи файлов. TFTP узко специализируется на выполнении двух операций передачи файлов: чтение и запись. Протокол TFTP не очень быстр и устойчив, но весьма прост в реализации (эти два критерия его разработки заявлены в спецификации RFC 783). В RFC 906 «Загрузка при помощи TFTP» (*Bootstrap Loading using TFTP*, Finlayson, 1984) TFTP заявлен как стандарт Интернет для выполнения задачи копирования загрузчиков по сети для бездисковых рабочих станций. Особенно полезен для инициализации сетевых устройств – маршрутизаторы, мосты, концентраторы.

Протокол TFTP, в отличие от FTP, не содержит возможностей аутентификации (хотя возможна фильтрация по IP-адресу) и основан на транспортном протоколе UDP. Чтобы гарантировать доставку данных между TFTP-сервером и TFTP-клиентом TFTP использует систему подтверждений [9].

#### **а) Сценарий TFTP**

Действие TFTP начинается UDP-датаграммой - запросом на чтение или на запись файла. Клиент начинает работу после получения порта, соответственно, посылая запрос на чтение или на запись на порт 69.

Сервер идентифицирует различные номера портов клиентов и использует их для последующей пересылки данных. Он направляет свои сообщения на порт клиента, посылая требуемый файл блоками длиной 512 байт. Сервер ждет подтверждения принятия клиентом каждого блока данных прежде, чем передать следующий. В качестве сигнала завершения передачи TFTP-сервер посылает UDP-датаграмму с длиной меньше 512 байтов. Если размер пересылаемого файла кратен 512, то признак EOF должен посылаться дополнительно. Таким образом, пересылка данных производится как обмен блоками данных и сообщениями АСК. Блоки данных нумеруются от единицы. Каждый АСК содержит номер блока данных, получение которого он подтверждает.

В улучшенном варианте TFTP разрешается согласование параметров через предварительные запросы чтения и записи. Его основная цель – позволить клиенту и серверу согласовывать между собой размер блока (более 512 байт) для увеличения эффективности пересылки данных.

### **б) Элементы данных протокола TFTP**

В TFTP существуют 5 типов элементов данных:

- запрос чтения (Read Request);
- запрос записи (Write Request);
- данные (Data);
- подтверждение (Acknowledgment);
- ошибка (Error).

Сообщение об ошибке указывает на события, подобные таким: «файл не найден», «файл не открыт», «на диске нет места».

Каждый заголовок начинается операционным кодом, который характеризует тип элемента данных протокола PDU (Protocol Data Unit). Форматы PDU представлены на рисунке 2.2.

Длина Read Request и Write Request меняется в зависимости от длины имени файла и полей режима, каждое из которых представляет собой текстовую строку ASCII, завершаемую нулевым байтом.

### **2.1.3. Протокол передачи файлов SFTP**

FTP использует два TCP-соединения для выполнения операций по передаче файлов; одно соединение для передачи команд и другое для передачи собственно данных. С другой стороны, TFTP использует для передачи файлов UDP-датаграммы. Протокол SFTP (*Simple File Transfer Protocol*, RFC 913, Lottor, 1984) является попыткой найти золотую середину между FTP и TFTP. SFTP поддерживает проверку прав пользователя (контроль доступа), передачу файлов, списки содержимого каталогов, изменение каталогов, переименование и удаление файлов. Подобно FTP, SFTP использует TCP. Однако в отличие от FTP, SFTP

использует только одно TCP-соединение. Команды, коды возврата и эксплуатационные особенности SFTP и FTP сильно схожи.

<b>Read Request</b> 2 октета	Строка	1 октет	Строка	1 октет
Операционный код =1	Имя файла	0	Режим	0

<b>Write Request</b> 2 октета	Строка	1 октет	Строка	1 октет
Операционный код =2	Имя файла	0	Режим	0

<b>Data</b> 2 октета	2 октета	
Операционный код =3	Номер блока	Данные

<b>Acknowledgment</b> 2 октета	2 октета
Операционный код =4	Номер блока

<b>Error</b> 2 октета	2 октета	Строка	1 октет
Операционный код =5	Код ошибки	Сообщение об ошибке	0

Рисунок 2.2 – Форматы элементов данных TFTP

## 2.2 TELNET как технология удаленного доступа к ресурсам сети

TELNET (Teletype Network) — сетевой протокол для реализации текстового интерфейса по сети - это одна из самых старых информационных технологий Интернет. Исторически технология TELNET служила для удалённого доступа к интерфейсу командной строки

операционных систем. Впоследствии ее стали использовать для прочих текстовых интерфейсов, вплоть до игр MUD и анимированного ASCII-art.

В настоящее время под TELNET понимают триаду, состоящую из telnet-интерфейса пользователя, telnetd-процесса, TELNET-протокола.

Эта триада обеспечивает описание и реализацию сетевого терминала для доступа к ресурсам удаленного компьютера. В настоящее время существует достаточно большое количество программ, которые позволяют работать в режиме удаленного терминала, но ни одна из них не может сравниться с TELNET по степени проработанности деталей и концепции реализации.

### 2.2.1 Протокол TELNET

TELNET как протокол описан в RFC-854. Его авторы J.Postel и J.Reynolds во введении к документу определяют назначение TELNET следующим образом: "...дать общее описание, насколько это только возможно, двунаправленного, восьмибитового взаимодействия, главной целью которого является обеспечение стандартного метода взаимодействия терминального устройства и терминал-ориентированного процесса. При этом протокол может быть использован и для организации взаимодействий "терминал-терминал" (связь) и "процесс-процесс" (распределенные вычисления)". Теоретически, даже обе стороны протокола могут являться программами, а не человеком. TELNET строится как протокол приложения над транспортным протоколом TCP. В основу TELNET положены три фундаментальные идеи:

- концепция сетевого виртуального терминала NVT (Network Virtual Terminal) ;
- принцип договорных опций (согласование параметров взаимодействия);
- симметрия связи "терминал-процесс".

Хотя в сессии Telnet выделяют клиентскую и серверную сторону, протокол на самом деле полностью симметричен. Симметрия

взаимодействия позволяет в течении одной сессии программе «user» и программе "server" меняться местами. Это принципиально отличает взаимодействие в рамках telnet от традиционной модели клиент-сервер. Терминальная программа ("user") и процесс ("server"), работающий с ней, преобразуют характеристики физических устройств в спецификацию NVT. Это позволяет, с одной стороны, унифицировать характеристики физических устройств, а с другой - обеспечить принцип совместимости устройств с разными возможностями. Характеристики диалога диктуются устройством с меньшими возможностями. Если взаимодействие осуществляется по принципу "терминал-терминал" или "процесс-процесс", то "user" - это сторона, иницирующая соединение, а "server" - пассивная сторона.

После установления транспортного соединения (как правило, TCP) оба его конца играют роль сетевых виртуальных терминалов, обменивающихся двумя типами данных:

- прикладными данными (т.е. данными, которые идут от пользователя к текстовому приложению на стороне сервера и обратно);
- опциями протокола Telnet, служащими для уяснения возможностей и предпочтений сторон.

Прикладные данные проходят через протокол без изменений [1], т.е. на выходе второго виртуального терминала видно именно то, что было введено на вход первого. С точки зрения протокола данные представляют просто последовательность байтов (октетов), по умолчанию принадлежащих набору ASCII, но при включенной опции Binary - любых. Хотя были предложены расширения для идентификации набора символов [10], но на практике ими не пользуются.

При установке telnet-соединения программа, работающая с реальным терминальным устройством, и процесс обслуживания этой программы используют для обмена информацией спецификацию представления правил функционирования терминального устройства или NVT. *NVT* - это

стандартное описание наиболее широко используемых возможностей реальных физических терминальных устройств. NVT позволяет описать и преобразовать в стандартную форму способы отображения и ввода информации.

Следует иметь в виду, что сессия Telnet уязвима для любого вида атак, к которым уязвим транспорт, т.е. протокол TCP, если только не осуществляется в полностью контролируемой сети или с применением защиты на сетевом уровне (различные реализации виртуальных частных сетей). Это связано с тем, что в протоколе не предусмотрено использование ни шифрования, ни проверки подлинности данных. Для функциональности удалённого доступа к системе в настоящее время применяется сетевой протокол SSH, при создании которого значительное внимание было уделено вопросам безопасности.

### **2.2.2 Организация сеанса работы с Telnet**

С помощью программы-клиента Telnet можно войти в систему удаленного компьютера как если бы ваш компьютер был терминалом того. Необходимо сообщить клиенту имя (адрес) удаленного компьютера, к которому необходимо присоединиться. Клиент, в свою очередь, устанавливает TCP-соединение с портом 23 (официальный порт TELNET). Удаленный компьютер показывает приглашение для входа в систему, в котором можно ввести имя пользователя. В зависимости от сервера Telnet, с которым установлено соединение, можно также получить приглашение для ввода пароля. Подобно анонимным FTP-серверам в Интернет существует большое количество общедоступных серверов Telnet. Различие между ними состоит в том, что серверы Telnet не имеют стандартного имени или пароля для анонимного доступа.

Клиент Telnet может устанавливать TCP-соединение с любым портом протокола фактически на любом сетевом компьютере Интернет. Эту способность клиента Telnet можно использовать для проверки и

тестирования работы других протоколов, Finger (порт 79), POP3 (порт 110), SMTP (порт 25), например: *telnet host.domain.org 25*.

### **2.2.3. Программа-клиент telnet**

Telnet - это интерфейс пользователя для работы по протоколу TELNET. Программа работает в двух режимах: в режиме командной строки (command mode) и в режиме удаленного терминала (input mode).

При работе в режиме удаленного терминала telnet позволяет работать с буферизацией (line-by-line) или без нее (character-at-a-time). При работе без буферизации каждый введенный символ немедленно отправляется на удаленную машину, откуда приходит "эхо". При буферизованном обмене введенные символы накапливаются в локальном буфере и отправляются на удаленную машину пакетом. "Эхо" в последнем случае также локальное. Для переключения между режимом командной строки и режимом терминала используют последовательность, которая может быть изменена командами TELNET, приведенными в [5].

### **2.2.4. Программа-сервер (telnetd)**

Telnetd - это сервер, который обслуживает протокол TELNET. Обычно telnetd запускается через сервис Интернет (inetd), но в некоторых системах может быть запущен и вручную. Telnetd обслуживает TCP-порт 23, но может быть запущен и на другой порт. В процессе работы сервер "слушает" порт TCP. В случае поступления запроса на обслуживание, telnetd назначает каждому удаленному клиенту псевдотерминал (pty) в качестве стандартного файла ввода (stdin), стандартного файла вывода (stdout) и стандартного файла ошибок (stderr).

При установке соединения с удаленным клиентом telnetd обменивается командами настройки (эхо, обмен двоичной информацией, тип терминала, скорость обмена, переменные окружения).

## **2.3 Технологии отложенного просмотра**

При чтении обзоров по технологиям Интернет можно обнаружить разделение систем по принципам функционального назначения,

существовавшим в эпоху больших универсальных ЭВМ, а именно: системы коммуникаций «пользователь-пользователь» (или системы информационного обмена) и системы коммуникаций – «пользователь-процесс» (или системы информационного обеспечения). Такая классификация имеет своей целью упорядочить представление о роли и месте хорошо известных информационных технологий как электронная почта (e-mail) и группа новостей (Newsgroup). К технологиям отложенного просмотра можно отнести e-mail, BBS, Usenet, Listserver. Классификацию технологий можно провести по виду систем коммуникаций, в которых они используются. Системы отложенного просмотра реализуют следующий алгоритм взаимодействия пользователей - передача и просмотр сообщений разделены во времени.

### **2.3.1 Технология систем индивидуальных коммуникаций**

Электронную почту можно рассматривать как технологию отложенного просмотра систем индивидуальных коммуникаций. Термин e-mail объединяет протоколы передачи данных и программное обеспечение, позволяющее пользователям компьютерной сети обмениваться почтовыми сообщениями (письмами). Технология сетевой почты, разработанная в рамках программы APRANET, коренным образом изменила способы взаимного общения, как в общественной, так и в частной сфере. В докладе, подготовленном ARPA в 1976 году, говорилось: «Служба отправки сообщений является неожиданным и незапланированным аспектом сети. Ее зарождение представляется больше похожим на открытие некоего природного явления, а не на целенаправленную разработку новой технологии».

Концепция электронной почты, ее основные компоненты подробно рассмотрены в [5]. В данном пособии остановимся на протоколе IMAP (Internet Mail Access Protocol) - интернет-протоколе прикладного уровня для доступа к электронной почте.

Напомним, что протокол SMTP используется для отправки почты от пользователей к серверам и между серверами для дальнейшей пересылки к получателю. Данные передаются при помощи TCP, при этом обычно используется порт 25 или 587. При передаче сообщений между серверами обычно используется порт 25. Для приёма почты почтовый клиент должен использовать протоколы POP3 или IMAP. Стандарт протокола POP3 определён в RFC 1939. Расширения и методы авторизации определены в RFC 2195, RFC 2449, RFC 1734, RFC 2222, RFC 3206, RFC 2595. Существуют реализации POP3-серверов, поддерживающие TLS и SSL.

Альтернативным протоколом для сбора сообщений с почтового сервера является IMAP. Главным отличием от POP является возможность поиска нужного сообщения и разбор заголовков сообщения. Как и POP3, протокол IMAP использует концепцию клиент-сервер с набором команд, предоставляя пользователю возможности для работы с почтовыми ящиками, находящимися на центральном сервере. Почтовая программа, использующая этот протокол, получает доступ к хранилищу корреспонденции на сервере так, как будто эта корреспонденция расположена на компьютере получателя. Электронными письмами можно манипулировать с компьютера пользователя (клиента) без необходимости постоянной пересылки с сервера и обратно файлов с полным содержанием писем.

IMAP был разработан для замены более простого протокола POP3 и имеет следующие преимущества:

- письма хранятся на стороне сервера, а не на стороне клиента. Возможен доступ к одному и тому же почтовому ящику с разных клиентов. Поддерживается также одновременный доступ нескольких клиентов. В протоколе есть механизмы, с помощью которых клиент может быть проинформирован об изменениях, сделанных другими клиентами;

- поддержка нескольких почтовых ящиков (или папок). Клиент может создавать, удалять и переименовывать почтовые ящики на сервере, а также перемещать письма из одного почтового ящика в другой;

- возможно создание общих папок, к которым могут иметь доступ несколько пользователей;

- информация о состоянии писем хранится на сервере и доступна всем клиентам. Письма могут быть помечены как прочитанные, важные и т. п.;

- поддержка поиска на сервере. Нет необходимости скачивать с сервера множество сообщений, для того чтобы найти одно нужное;

- поддержка онлайн-работы. Клиент может поддерживать с сервером постоянное соединение, при этом сервер в реальном времени информирует клиента об изменениях в почтовых ящиках, в том числе о новых письмах;

- предусмотрен механизм расширения возможностей протокола.

Текущая версия протокола имеет обозначение IMAP4rev1 (IMAP, версия 4, реализация 1). IMAP-сервер использует TCP-порт 143 или 993 (IMAP поверх SSL).

На рисунке 2.3 представлена схема состояний для протокола IMAP [7], где цифрами обозначены следующие действия:

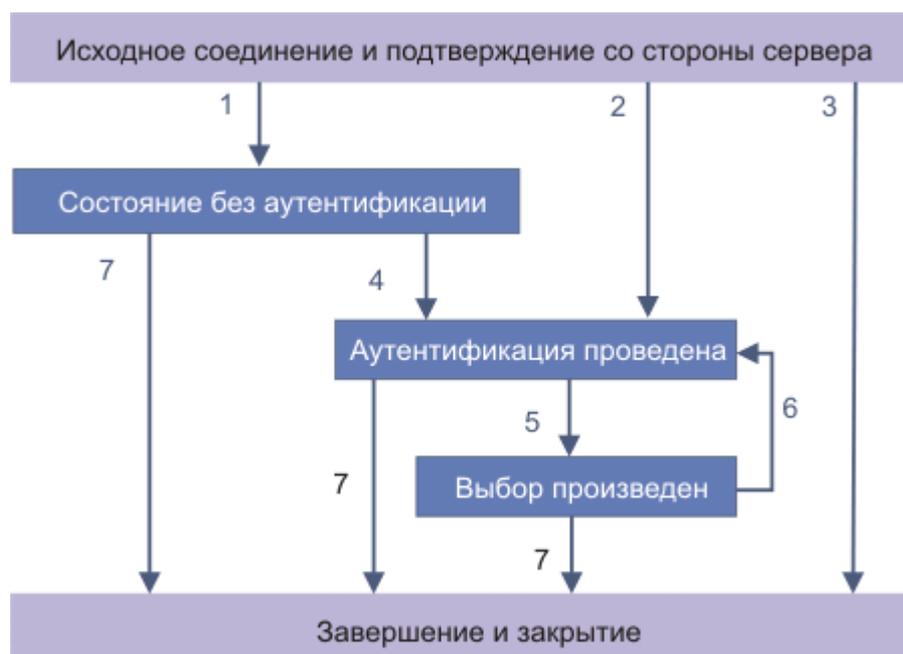


Рисунок 2.3 - Схема состояний для протокола IMAP

- 1 - соединение без предварительной аутентификации (отклик OK);
- 2 - соединение с предварительной аутентификацией (отклик PREAUTH);
- 3 - соединение отвергнуто (отклик BYE);
- 4- успешное завершение команды LOGIN или AUTHENTICATE;
- 5 - успешное завершение команды SELECT или EXAMINE;
- 6- выполнение команды CLOSE, или неудачная команда SELECT или EXAMINE;
- 7 -выполнение команды LOGOUT, закрытие сервера, или прерывание соединения.

Сервер IMAP 4.1 может находиться в одном из четырех состояний. Большинство команд допустимо только во вполне определенных состояниях. Если клиент пытается реализовать команду в неправильном состоянии, это рассматривается как протокольная ошибка. В этом случае сервер откликнется командой *bad* или *no* в зависимости от реализации конкретной программы.

В состоянии без аутентификации клиент должен предоставить имя и пароль, прежде чем станет доступно большинство команд. Переход в это состояние производится при установлении соединения, если только для данного соединения не была проведена предварительная аутентификация.

В состоянии аутентификации клиент идентифицирован и должен выбрать почтовый ящик, прежде чем ему станут доступны команды для работы с сообщениями. Переход в это состояние происходит при установлении соединения с предварительной аутентификацией, когда выданы все необходимые идентификационные данные или при ошибочном выборе почтового ящика.

В состояние выбора система попадает, когда успешно осуществлен выбор почтового ящика. В состояние выхода система попадает при прерывании соединения в результате запроса клиента или вследствие независимого решения сервера.

Данные в IMAP 4.1 могут иметь одну из следующих форм: атом, число, строка, список, заключенный в скобки или NIL. *Атом* состоит из одного или более неспециализированных символов. *Число* состоит из одной или более цифр и характеризует некоторое числовое значение. *Строка* может иметь одну из двух форм: литерал или строка в кавычках. Литеральная форма является основной формой строки. Строка в кавычках является альтернативной формой, исключающей избыточность литеральной формы за счет ограничений, налагаемых на символы, используемые в строке. *Литерал* представляет собой нуль или более октетов (включая CR и LF). Литерал начинается с октета, где хранится число символов. Этот октет заключается в фигурные скобки, за которыми следует последовательность CRLF. В случае передачи литералов от сервера к клиенту за CRLF следуют непосредственно данные. При передаче литералов от клиента серверу клиент должен подождать прихода команды продолжения, прежде чем начать пересылку данных. *Строка в кавычках* представляет собой последовательность из нуля или более 7-битовых символов за исключением CR и LF, начинающуюся и завершающуюся двойной кавычкой ". Пустая строка представляется как "" или как литерал {0}, за которым следует последовательность CRLF. Если число октетов равно нулю, клиент, передающий литерал должен подождать прихода команды продолжения.

### 2.3.2 Технологии систем коллективных коммуникаций

К технологиям отложенного просмотра систем коллективных коммуникаций можно отнести списки рассылки и группу новостей Usenet.

Под термином «списки рассылки» (listserv, majordomo, almanac) понимается централизованная система, имеющая один "центральный" узел, который занимается сбором и формированием статей/файлов для рассылки. Поместить статью/файл на сервер рассылки или получить оттуда статью/файл пользователь может только одним способом -

обратившись напрямую к этому серверу. Обычно все списки рассылки модерируемы.

Группа новостей (телеконференции) представляет собой довольно сложную распределенную систему, состоящую из узлов, каждый из которых подписан на определенную телеконференцию, где есть полный набор опубликованных статей. Пользователь может заказать статью со своего узла или с любого ближайшего, который имеет эту конференцию. Узлы же постоянно обмениваются статьями "своих" пользователей. В результате последней реорганизации все группы Usenet были разбиты на несколько основных категорий:

- alt.\* - разнородная информация,
- comp.\* - информация о компьютерах,
- news.\* - актуальные новости, связанные с Usenet,
- rec.\* - отдых и увлечения,
- sci.\* - научные дискуссии,
- soc.\* - социальные вопросы,
- misc.\* - темы, не относящиеся к другим категориям.

Принцип классификации Usenet отличается от классификации доменов, например, для опубликования новостей отдельных организаций или региональных групп новостей, создаются отдельные иерархии, например:

- netcom.\* - для фирмы Netcom Communications Service,
- ufl.\* - для университета штата Флорида,
- dc.\* - для округа Колумбия.

Количество иерархий в группе растет, например, категория comp (все, что связано с компьютерами), прибавляем к имени темы sys (от system), получаем темы, связанные с конкретными компьютерными системами. Добавляем ibm.pc, получаем comp.sys.ibm.pc – т.е. иерархию группы новостей, в которой обсуждается все, что связано с компьютерами IBMPC.

Сообщение оформляется в соответствии со стандартом почтового сообщения Интернет (RFC-822). Фактически Usenet выполняет функции периодического издания. Пользователь может подписаться на любом сервере на любую группу новостей, Каждый сервер ведет список подписчиков и список телеконференций, на которые осуществлена подписка. При поступлении статьи, сервер информирует известные ему серверы о том, что появилась новая информация и в случае запроса передает ее. Это взаимодействие осуществляется посредством протокола обмена новостями -NNTP.

Протокол NNTP пришел на смену UUCP, его целью было упорядочить обмен информацией между серверами Usenet. Кроме того, использование интерактивного протокола позволило разработать интерфейсные программы-клиенты, позволяющие просматривать почту на удаленных серверах.

Протокол NNTP, как и SMTP, является текстовым, то есть все команды и ответы на них являются обычными текстовыми строками. Важной особенностью протокола NNTP является его эффективность в случае сложных графов связей между серверами новостей. Чтобы одно и то же сообщение не передавалось многократно, обычно отправляющий сервер сначала сообщает идентификатор нового сообщения, а само сообщение отправляет только после подтверждения принимающей стороны о том, что этого сообщения там ещё нет. В случае расширения stream NNTP эта концепция развивается ещё дальше: отправляющая сторона шлёт список идентификаторов новых статей, не дожидаясь ответа на каждый из них, и сами статьи, если принимающая запросила их по идентификаторам.

Протокол NNTP определяет запросно-ответный механизм обмена сообщениями между серверами и между сервером и программами-клиентами. Для этой цели в протоколе определен набор команд и ответов

на них. Весь диалог осуществляется в текстах ASCII. Каждая команда состоит из идентификатора команды и параметров.

Существует вариация протокола NNTP, называемая NNRP (Network News Readers Protocol). Она отличается только набором поддерживаемых команд, и предназначена для чтения конференций с сервера новостей клиентским ПО в режиме онлайн. Функционально NNTP ориентирован на то, что статьи отправляются подписчикам при их появлении на сервере, а NNRP — по запросу клиента. За NNTP закреплен TCP-порт 119. При подключении к NNTP-серверу по SSL (т. н. NNTPS) используется порт 563.

Телеконференции и списки рассылки делятся на открытые и закрытые. *Открытые* конференции - для всех, *закрытые* конференции - для ограниченного числа пользователей.

По контролю конференции и списки рассылки можно разделить на группы: модерируемые, немодерируемые и пост-модерируемые.

*Модерируемые* (управляемые) - сначала направят Вашу заметку ведущему (модератору) и, если он сочтет статью достойной всеобщего внимания, то уже он направит ее в конференцию. *Немодерируемые* - каждый пишет, что хочет, соблюдая устав и тему данной конференции. *Пост-модерируемые* (наблюдаемые) - поместить статью может любой, но есть человек, который следит, чтобы все уже опубликованные статьи соответствовали теме и уставу конференции.

### **Контрольные вопросы**

- 1) Назовите базовые технологии Интернет.
- 2) Назовите цель и средства базовых технологий Интернет.
- 3) Каковы особенности технологии взаимодействия с интерактивным пользователем?
- 4) Назовите основные этапы работы в FTP.
- 5) Какие команды FTP используются для передачи файла?

- 6) Назовите опции FTP для операций по передаче файла.
- 7) Что означает тип файла в FTP?
- 8) Что такое формат файла в FTP?
- 9) Как FTP использует протокол TELNET на управляющем соединении?
- 10) Каково соотношение FTP и TELNET ?
- 11) Как протестировать соединение?

### **РАЗДЕЛ 3. WEB-ТЕХНОЛОГИИ**

#### **3.1 Основные компоненты Web-технологии**

Под Web-технологиями понимается подмножество Интернет-технологий, упрощающих доступ к мультимедиа информации. Сильные стороны Web-технологий - в первую очередь, это простота применения и независимость от операционной системы. Однако из всего разнообразия решений, доступных сегодня, можно выделить те основные технологии, которые применяются для создания практически любых сайтов или других Интернет-проектов (рисунок 3.1):

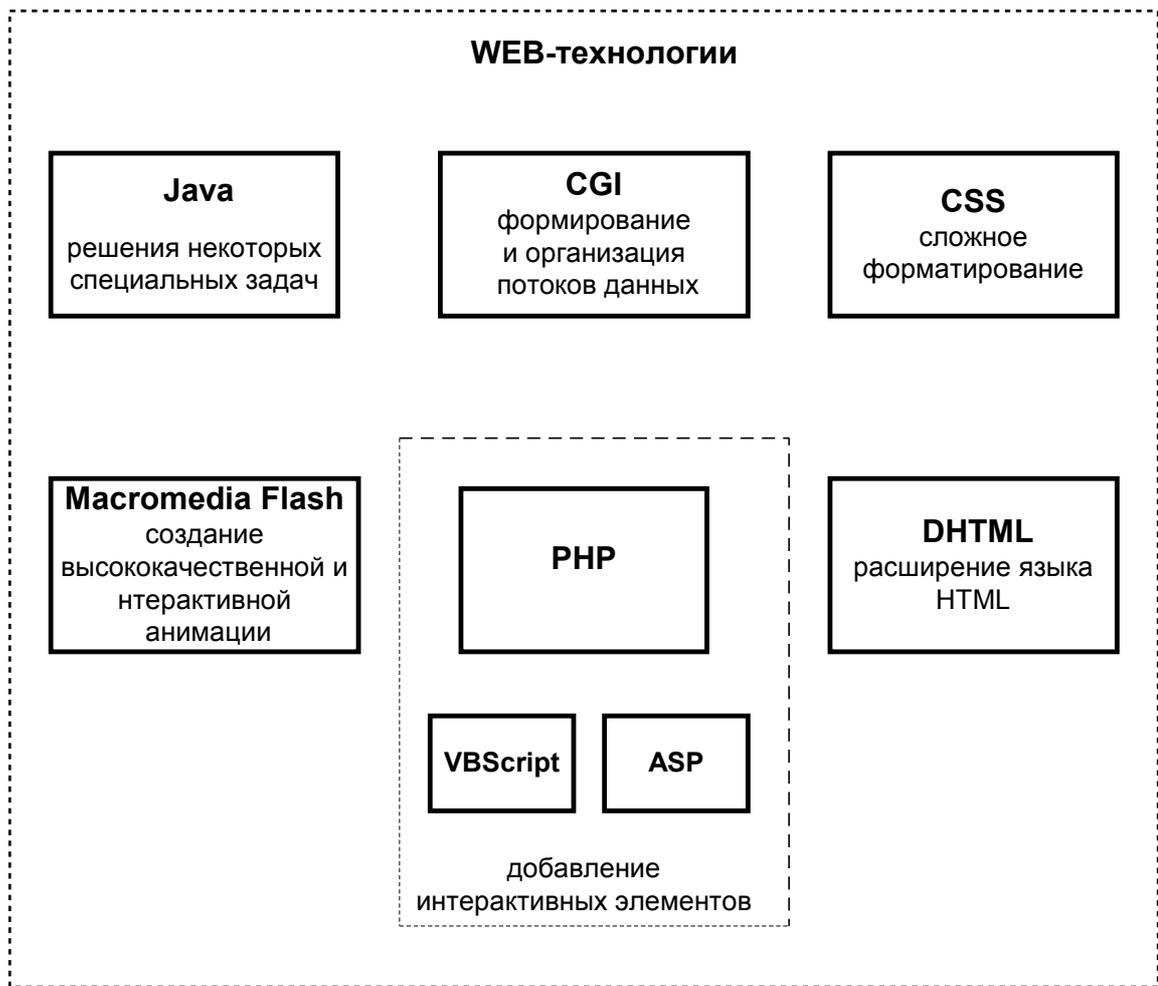


Рисунок 3.1 – Основные Web-технологии

- Java - язык программирования, в настоящее время используется для решения некоторых специальных задач. Java позволяет создавать многочисленные активные элементы, придавать Интернет-странице большую интерактивность;

- CGI (Common Gateway Interface) - большое количество Web-приложений основано на использовании внешних программ, управляемых Web-сервером. Для связи между Web-сервером и вызываемыми программами широко используется CGI, имеющий реализации как для Windows-ориентированных программ, так и для приложений, функционирующих в среде Unix. Данный документ описывает Windows-модификацию интерфейса CGI. Сайты, основанные на использовании данной технологии позволяют формировать и организовывать потоки данных. Использование данных программ позволяет строить Web

приложения с динамически обновляемой информацией, хранящейся в базах данных или генерирующейся в зависимости от бизнес-правил решаемых задач. Именно на основе CGI построено большинство досок объявлений, форумов, чатов и конференций;

- CSS (Cascading Style Sheets) - технологии такого типа позволяют осуществлять сложное форматирование, необходимое при создании практически любого сайта. Таблицы каскадных стилей, содержащие набор свойств для описания внешнего вида любых HTML-документов, позволяют иметь полный контроль над стилем и расположением каждого элемента web-страницы, что проще и гораздо функциональнее использования обычного набора HTML тегов;

- PHP (PHP: Hypertext Preprocessor) - встраиваемый в HTML язык программирования сценариев. В настоящее время нашел наибольшее применение для разработки интерактивных Web-приложений. Сайты, построенные на основе PHP позволяют добавлять на web-страницы различные интерактивные элементы. Сюда же можно отнести web-технологии на основе VBScript и ASP. Все они предназначены для придания страницам большей интерактивности;

- Macromedia Flash - технология веб-мультипликации и создания интерактивного контента компании от Macromedia, получившая широкое распространение. Основное назначение этой web-технологии – создание высококачественной интерактивной анимации. Файлы в формате Flash чрезвычайно компактны, однако создание их весьма трудоемко;

- DHTML- специальное расширение языка HTML. Web-технологии DHTML позволяют включать разнообразные интерактивные элементы. Например, движущийся фон или объекты, выпадающие меню, бегущие титры и т.д.

Применение Web-технологий позволяет значительно упростить задачу обеспечения совместимости при разработке приложений

управления и, возможно, может послужить основой для создания интерфейса с хранилищами данных [8].

### 3.2 Использование модели «клиент-сервер» для WWW

WWW построена по модели «клиент-сервер». На рисунке 3.2 показано разделение функции в этой известной модели.

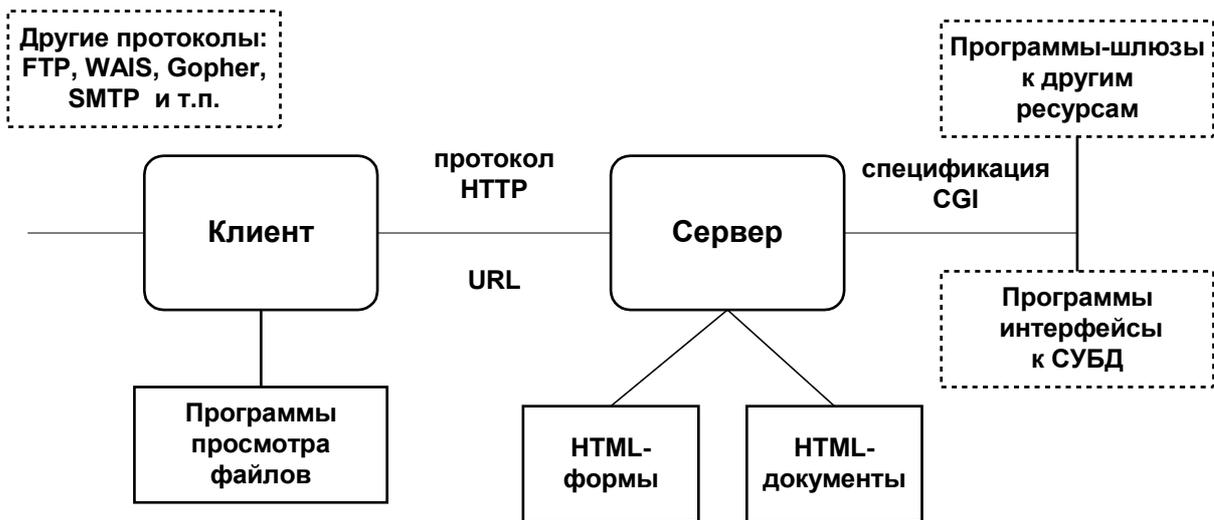


Рисунок 3.2 – Модель клиент - сервер

Программа-клиент выполняет функции интерфейса пользователя и обеспечивает доступ практически ко всем информационным ресурсам Интернет. В этом смысле она выходит за обычные рамки работы клиента с сервером только определенного протокола, как это происходит, например, в Telnet. Фактически, клиент - это интерпретатор HTML, и как типичный интерпретатор, клиент в зависимости от команд (разметки) выполняет различные функции. В круг этих функций входит не только размещение текста на экране, но обмен информацией с сервером по мере анализа полученного HTML-текста. Наиболее наглядно это происходит при отображении встроенных в текст графических образов. При анализе URL-спецификации или по командам сервера клиент запускает дополнительные внешние программы для работы с документами в форматах, отличных от HTML, например GIF, JPEG, MPEG, Postscript и т.п. В последнее время

большее распространение получил механизм согласования запускаемых программ через MIME-типы.

Другую часть программного комплекса WWW образует сервер протокола HTTP, базы данных документов в формате HTML, управляемые сервером, и программное обеспечение, разработанное в стандарте спецификации CGI. До образования Netscape реально использовалось два HTTP-сервера: сервер CERN и сервер NCSA. В настоящее время число базовых серверов расширилось, например, сервер для MS-Windows и Apache-сервер для Unix-платформ. Существуют и другие, но два последних можно выделить из соображений доступности использования. Сервер для Windows - это Shareware, но без встроенного самоликвидатора, как в Netscape. Разработанный Apache является Freeware. Он реализует также новые дополнения к протоколу HTTP, связанные с защитой от несанкционированного доступа, которые предложены группой по разработке этого протокола и используются практически во всех коммерческих серверах.

База данных HTML-документов - это часть файловой системы, которая содержит текстовые файлы в формате HTML и связанные с ними графику и другие ресурсы. Особое внимание хотелось бы обратить на документы, содержащие элементы экранных форм. Эти документы реально обеспечивают доступ к внешнему программному обеспечению.

Прикладное программное обеспечение, работающее с сервером, можно разделить на программы-шлюзы и прочие. Шлюзы - это программы, обеспечивающие взаимодействие сервера с серверами других протоколов, например ftp, или с распределенными на сети серверами Oracle. Прочие программы - это программы, принимающие данные от сервера и выполняющие какие-либо действия: получение текущей даты, реализацию графических ссылок, доступ к локальным базам данных или просто расчеты.

### **3.3 Основные компоненты технологии Active-X**

Active-X - собирательный термин технологий Microsoft, которые позволяют разработчикам создавать интерактивные приложения для Интернет. Длительное время в Интернет преобладали статические страницы, чтение которых было утомительным занятием. Затем на серверах стали появляться элементы программирования кнопок, двигающихся пиктограмм, графических и видеоменю, элементов виртуальной реальности, видеоконференций, средств визуального поиска и авторизованного доступа. Чтобы помочь программистам в написании всех этих программ и стандартизировать сам подход к их написанию, была создана платформа Active-X. Эта платформа базируется на COM (Component Object Model). Технология включает следующие основные компоненты:

- Active-X Controls - позволяет запускать анимацию, видео, аудио без дополнительных программ;
- Active-X Scripting - обеспечивает поддержку скриптов;
- Active-X Documents - позволяет запускать программы для просмотра документов, отличных от HTML, например, Word, Excel.

В общем случае, Active-X - это набор API, реализованный как для клиента, так и для сервера. Клиентская часть состоит из следующих компонент:

- программных компонент Active-X (OLE controls) - передаются в составе документа HTML и обеспечивают взаимодействие с пользователем;
- виртуальной машины Java - позволяет работать программным компонентам (applets) Java и взаимодействовать с компонентами Active-X;
- активных документов Active-X - позволяют просматривать обычные документы (например, таблицы Excel или документы Word), используя механизм OLE Automation;
- Active Scripting - обеспечивает взаимодействие нескольких программных компонент Active-X или Java (например, синхронизацию звуко- и видеоряда).

Серверная часть включает в себя:

- интерфейс программирования ISAPI для взаимодействия сервера Web с другими приложениями;
- стандарт DCOM (Distributed Component Object Model) - позволяет напрямую взаимодействовать объектам на клиенте и сервере;
- сервисы OLE для просмотра проводником Internet-документов в формате, отличном от HTML.

### **3.4 Технология PHP**

В основе данной технологии лежит язык PHP (PHP: препроцессор гипертекста») - скриптовый язык программирования, созданный для генерации HTML-страниц на веб-сервере и работы с базами данных. Наряду с JSP, Perl и языками, используемыми в ASP.NET, PHP входит в LAMP — «стандартный» набор для создания веб-сайтов (Linux, Apache, MySQL, PHP (Python или Perl)).

Изначально разработанный для взаимодействия с HTML может заменять его в любой пропорции, от вставки нескольких строчек, до написания всего текста страницы на этом языке. Данный язык отличается простотой, скоростью выполнения, богатой функциональностью и распространением исходных кодов на основе лицензии PHP. Кроме того, PHP отличается наличием ядра и подключаемых модулей, «расширений»: для работы с базами данных, сокетами, динамической графикой, криптографическими библиотеками, документами формата PDF и т. п.. Немаловажно то, что PHP существует лишь на сервере, а на клиентскую машину он является в виде HTML-кода. Интерпретатор PHP подключается к веб-серверу либо через модуль, созданный специально для этого сервера (например, для Apache или IIS), либо в качестве CGI-приложения.

PHP может использоваться для решения административных задач в операционных системах UNIX, GNU/Linux, Microsoft Windows, Mac OS X и AmigaOS. Однако в таком качестве он не получил распространение, отдавая пальму первенства Perl, Python и VBScript.

Синтаксис PHP заимствован из таких языков как C, Java, Perl. Кроме того, этот язык существует в связке с Apache и SQL.

Для написания PHP-сценариев можно использовать любой редактор, при этом следует дать файлу расширение .php.

Для PHP существует четыре способа отделения его от общего кода HTML:

1) `<? echo ( "SGML инструкции | n" ); ?>`

2) `<?php echo ( "XML документ | n" ); ?>`

3) `<script language= "php">`  
`echo ("специально для FrontPage");`  
`</script>`

4) `<% echo ( "ASP-стиль" ); %>`

`<% = $variable; # комментарий "echo ( "ASP-стиль" ); %>`

Следует сказать, что инструкции в PHP отделяются друг от друга точкой с запятой (;), хотя перед закрывающимся тегом (?>) ставить точку с запятой (;) не обязательно.

В PHP, как и во многих языках программирования, существует средство для хранения данных – переменная. Для выбора имени переменной в PHP существует ряд синтаксических правил:

- любое имя переменной должно начинаться со знака доллара (\$);
- после него может идти либо буква, либо знак подчеркивания (\_), но не цифра;
- далее могут следовать буквы, цифры и символы подчеркивания в любой последовательности. Знак пробела недопустим!
- имена переменных чувствительны к регистру (\$lata и \$Lata не будут эквиваленты).

Кроме того, в соответствии с Coding Standart, при выборе имени переменной используются символы в нижнем регистре, слова разделяются знаками подчеркивания.

PHP поддерживает четыре скалярных (Integer, Double, Boolean, String) и два смешанных (Array, Object) типа данных. Помимо скалярных и смешанных, PHP поддерживает два специальных (resource, NULL) типа данных, а также несколько псевдотипов (mixed, number, callback).

В отличие от других языков программирования, в PHP не обязательно задавать тип данных, так как он автоматически определяется, когда присваивается значение.

Кроме того, в одной программе переменная может быть, например, числом и строкой. Для отслеживания текущего типа данных применяется функция *gettype()*. В круглых скобках указывается имя переменной, например, *echo gettype(\$var)*.

Если требуется проверка переменной на соответствие определенному типу данных, то применяются функции *is\_integer()*, *is\_double()*, *is\_string()*, *is\_array()*, *is\_object()* и *is\_bool()*. В случае соответствия типа переменной выводится 1, иначе - 0.

В PHP имеется возможность использовать переменные, имена которых содержат переменные, например:

```
<?php
$name = 'nata';           // $name содержит строку 'nata'
$id = 7;                 // $id содержит число 7
echo $$name              // выводит 7
?>
```

Такие переменные, как *\$\$name*, называют динамическими. Они применяются, как правило, если требуется в ходе выполнения программы создавать много переменных.

В PHP заранее определен ряд констант, которые называются предопределенными. Их основное назначение – хранить информацию о системе. Например, *PHP\_VERSION*, *PHP\_OS* содержат соответственно

версию PHP и название операционной системы, на которую установлен сервер.

Существует также несколько констант, которые могут менять свое значение в зависимости от их использования. Например, константы `__LINE__` и `__FILE__` содержат в себе соответственно номер строки и имя файла сценария. Эти константы можно писать как прописными буквами, так и заглавными.

Операторы классифицируются по количеству операндов, на которые они действуют. Обычно используются бинарные операторы, такие как сложение, вычитание и др. Они задействуют два операнда. Но в PHP есть и унарные операторы (используют один операнд), и тернарные (три операнда) [4].

Оператор цикла `foreach` является новым, и используется только начиная с PHP 4. Аналогичный оператор можно найти в Perl. Он призван упростить последовательную обработку всех элементов массива. Существуют две системы синтаксиса для этого оператора:

*Foreach (array\_expression as \$value) выражения*

*Foreach (array\_expression as \$key => \$value) выражения*

В первом случае значения переменных массива по очереди присваиваются переменной `$value`, над которой производятся действия. В итоге получается своеобразный указатель на то или иное значение переменной из необходимого массива.

Второй способ аналогичен первому, только в этом случае помимо значения переменной будет еще использоваться и ключ, т. е. обозначение этой переменной.

Массивы поддерживают числовые и строковые ключи и являются гетерогенными. Массивы могут содержать значения любых типов, включая другие массивы. Порядок элементов и их ключей сохраняется.

В общем виде синтаксис определения функции PHP следующий:

*function имя\_функции(арг1[=зн1], арг2[=зн2], ... аргN[=знN])*

*{ операторы\_тела\_функции; }*

Имя функции должно быть уникальным с точностью до регистра букв. Это означает, что, во-первых, имена `MyFunction`, `myfunction` и даже `MyFuNcTiOn` будут считаться одинаковыми, и, во-вторых, мы не можем переопределить уже определенную функцию (стандартную или нет — не важно), но зато можем давать функциям такие же имена, как и переменным в программе (конечно, без знака `$` в начале). Список аргументов, состоит из нескольких перечисленных через запятую переменных, каждую из которых необходимо задать при вызове функции (впрочем, когда для этой переменной присвоено через знак равенства значение по умолчанию (обозначенное `=zNM`), ее можно будет опустить. Конечно, если у функции не должно быть аргументов (как это сделано у функции `time()`), то следует оставить пустые скобки после ее имени, например:

```
function SimpleFunction() { ... }
```

В фигурные скобки заключается тело функции. В нем могут быть любые операторы, включая даже операторы определения других функций (правда, эти "другие функции" не будут локальными, а станут далее "видны" для всей программы, но только с того момента, как до их описания дойдет управление). Если функция должна возвращать какое-то значение, что среди них должен встретиться оператор `return`. Если же она должна отработать без возврата значений, то оператор `return` можно и не указывать (или указывать без задания возвращаемого значения).

Более сложным структурным компонентом программы (по сравнению с функцией) является объект. Для примера будем рассматривать класс объектов как расширение понятия функции. У функции в качестве аргументов были переменные, и то необязательно. У класса объектов в качестве аргументов могут быть тоже переменные (их обычно называют свойствами объекта класса). Но у класса объектов в качестве аргументов имеются и функции (их обычно называют методами). Упрощенно можно

сказать, что класс объектов — это связанная совокупность переменных и функций [2, 12].

В web-документах часто приходится применять различные формы для ввода пользователем данных. После щелчка на кнопке отправки данных данные отправляются в файл-сценарий и там обрабатываются. Рассмотрим простейшую форму с названием поля ввода, самим полем ввода и кнопкой для отправки данных. Класс таких объектов представлен ниже:

```
<?php
class input
{
    var $nam,$act;
function form_param($nam,$act)
    {
    $this-> nam = $nam;
    $this-> act = $act;
    }
function form_input()
{
echo "<form action=$this->act>
Input <input type='text' name=$this->nam>
    <input type='submit' value='Send'>
    </form>";
    }
}
$myinput = new input;
$myinput->form_param('user','script.php');
$myinput->form_input();
```

После ключевого слова class во второй строке стоит имя, которое дается этому классу, например, input. Далее в операторных скобках (третья и предпоследняя строки) следует собственно класс объектов простой формы ввода данных. В приведенном примере класс состоит из двух переменных и двух функций. Переменными обозначено имя поля ввода — \$nam и имя сценария, куда введенные данные будут отправляться для обработки — \$act. Имена переменных введены через запятую, после ключевого слова var. Первая функция form\_param() служит для приема значений переменных \$nam и \$act извне. Синтаксическое построение

`$this->nam = $nam` упрощенно можно перевести так: в этом классе переменной присваивается введенное значение переменной `$nam`. Аналогично и для второй строки.

Следующая функция `forminput()` генерирует в web-документе простую форму с именем поля ввода, равным значению `nam`. Action для формы будет равно значению `act`. Синтаксическое построение `$this->act` означает, что в качестве имени action берется значение переменной `act` этого же класса.

Для отладки сценариев применяется оператор подавления ошибки (`@`). Если поставить его перед выражением, то любые возникающие в нем ошибки или предупреждения, не будут выводиться в окне браузера:

```
<?php
@ $a = 5/0; // подавляет ошибку деления на 0
?>
```

При запуске `php`-файла автоматически инициализируются переменные с такими же именами, как у элементов формы. Для доступа к данным из HTML-формы используются суперглобальные массивы `$_GET` и `$_POST`. Использование того или иного массива зависит от метода передачи данных. В ниже приведенном примере выбирается метод передачи данных GET

```
<?php
echo $_GET['text']; // выводит значение поля text
?>
```

Если выбирается метод передачи данных GET, то создается элемент массива `$_GET`, ключ которого имеет название `text`. При использовании метода POST данные заносятся в массив `$_POST`.

Ссылки на внешние ресурсы представляются значениями типа «ресурс». Они, как правило, создаются и обрабатываются функциями из определённых расширений. Примерами ресурсов являются файлы, изображения, базы данных.

Пример программы на языке PHP, позволяющей узнать версию PHP-машины, установленной на компьютере или на сервере, услугами которого пользуетесь, необходимо запустить сценарий, приведенный ниже:

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

### **3.5 Технология CGI**

Технология CGI (Common Gateway Interface) подразумевает использование в составе ресурса Интернет интерактивных элементов на базе приложений, обеспечивающих передачу потока данных от объекта к объекту. Именно так организовано в Сети большинство конференций, досок объявлений, гостевых книг, поисковых машин и систем подсчета рейтинга [3].

Технология CGI обычно реализуется двумя методами: либо с использованием программ, написанных на языке PERL (Practical Extraction and ReportLanguage), — такие файлы имеют расширение .pl, либо с помощью приложений, созданных обычно с применением языка C и откомпилированных непосредственно на сервере, поскольку большинство UNIX-совместимых платформ включают в себя встроенный транслятор этого языка. Подобные программы имеют расширение .cgi. Необходимо отметить, что PERL, как и HTML, является интерпретируемым языком. Иными словами, для того чтобы запустить подобный скрипт на исполнение, не требуется дополнительной компиляции. Помимо упомянутых возможностей с помощью данной технологии можно организовать систему показа последовательности рекламных баннеров или автозагрузки файлов на сервер, создать форму отправки электронного письма непосредственно со страницы сайта или службу виртуальных открыток.

Среди достоинств CGI следует отметить их независимость от клиентского программного обеспечения — эту технологию сможет

применять каждый пользователь, просматривающий содержимое сервера при помощи браузера практически любой версии.

Основной недостаток заключается в том, что для установки и настройки приложений CGI на сервере нужно обладать как минимум правами администратора, поскольку эти программы при запуске способны нарушить нормальное функционирование серверного компьютера и дестабилизировать работу сети.

В общем случае принцип работы CGI выглядит следующим образом: пользователь заполняет на web-страничке ту или иную форму и нажимает на кнопку, после чего встроенная в код HTML-строка вызова CGI-скрипта запускает соответствующую программу CGI и передает ей управление процессом обработки информации. Введенные пользователем данные отсылаются этой программе, а она, в свою очередь, встраивает их в другую страницу, отправляет по почте или трансформирует каким-либо иным способом. Скрипты CGI размещаются на сервере в специально отведенной для этих целей директории, которой, как правило, назначается имя CGI-BIN.

CGI-программа возвращает результат работы, отвечающий (явно или неявно) целям запроса. Сервер кодирует результат работы в соответствии со стандартом HTTP и использует HTTP для отправки результата клиенту, т.е. сервер добавляет необходимые HTTP заголовки в сообщение, формируемое CGI - программой.

Результат работы CGI программы состоит из двух частей: заголовка и тела сообщения. Заголовок состоит из одной или более строк текста, отделенных от тела пустой строкой. Тело сообщения содержит данные, представленные в MIME формате, указанном в заголовке.

Сервер не изменяет тело документа, т.е. сервер передает сформированный CGI-программой ответ "как он есть". Сервер распознает следующие строки заголовка в выходном потоке:

1) Content-Type - указывает на MIME тип тела сообщения. Значение этого параметра должно быть в формате type/subtype.

*URI: <value> (value enclosed in angle brackets)*

Данное значение указывает на полный URL или ссылку на локальный файл, сообщение из которого будет возвращено клиенту в теле сообщения. Если значение является локальным файлом, сервер отсылает его как результат запроса, как будто клиент воспользовался методом GET при генерации запроса. Если значение является полным URL, то сервер возвращает сообщение "401 redirect" для обеспечения прямой загрузки указанного объекта.

2) Location - То же самое, что и URI, но данная форма сейчас не используется. Параметр value не должен быть взят в угловые скобки.

Другие заголовки передаются клиенту в том виде, в котором они представлены.

Сервер позволяет конечному приложению осуществлять прямой возврат результата запроса клиенту. Это осуществляется посредством включение в заголовок возвращаемого сообщения его информационного протокола, что позволяет CGI-программам формировать непосредственный ответ клиенту с указанием HTTP заголовка без предварительной обработки его сервером..

Сервер анализирует результат запроса, помещаемый CGI программой в выходной файл (Output File), и, если первая строка "HTTP/1.0", он предполагает, что сообщение содержит полный HTTP ответ и отсылает его клиенту без упаковки.

### **3.6 Технология Java**

Разработчики распределенных масштабируемых многопользовательских систем, реализующие бизнес-логику, всегда искали эффективные средства построения таких систем, решающих поставленные задачи и удовлетворяющих высоким требованиям, предъявляемым к подобным системам. Использование Java-технологии

позволяет быстро и эффективно создавать программное обеспечение уровня предприятия, масштабируемые, защищенные и надежные системы, допускающие повторное использование компонентов. Большинство web-камер, передающих на сайт живое изображение, также работают на базе соответствующих приложений Java.

Инструмент данной технологии, а именно язык программирования Java, разработан компанией Sun Microsystems и напоминает по структуре и синтаксису хорошо знакомый язык С. Язык Java существует сегодня в Интернете в двух вариантах: JavaScript и собственно Java.

Первый вариант языка является всего лишь надстройкой стандарта HTML и значительно расширяет возможности документа, созданного в этом формате. Модуль, написанный на JavaScript, интегрируется в файл HTML как подпрограмма и вызывается на исполнение из соответствующей строки HTML-кода стандартной командой. Встроенный в браузер интерпретатор языка воспринимает и скрипт, и сам код гипертекста как единый документ, обрабатывая те и другие данные одновременно.

Модуль Java в отличие от JavaScript не интегрируется в использующую его страницу, а существует как самостоятельное приложение (апплет) с расширением .class. При использовании этого варианта языка, апплет также вызывается из html-файла соответствующей командой, но загружается, инициализируется и запускается на исполнение в виде отдельной программы, в фоновом режиме.

Среди достоинств этой технологии следует отметить отсутствие необходимости устанавливать и настраивать на сервере какие-либо дополнительные модули, обеспечивающие работу Java-программ. Главный недостаток Java заключается в том, что пользователи браузеров старых версий, не поддерживающих компиляцию данного языка, воспринимать объекты, созданные при помощи Java и JavaScript, не смогут.

### **3.7 Технология CSS**

В ряде случаев у web-дизайнера возникает необходимость применить в процессе создания html-документа сложное форматирование - от абзаца к абзацу менять шрифт, расположение текста, его цвет, формировать различные таблицы данных.

Можно решить эту проблему с помощью стандартных средств HTML. В стандартном HTML для присвоения какому-либо элементу определенных свойств (таких, как цвет, размер, положение на странице и т. п.) необходимо каждый раз описывать эти свойства, даже если на одной страничке должны располагаться 2 или 20 таких элементов, ничуть не отличающихся один от другого, т.е. два или более раз вставить один и тот же кусок HTML-кода в страничку, увеличивая размер файла и время загрузки страницы.

Другой подход, основанный на применении технологии CSS, позволяет подключить к странице внешний файл, выполненный в стандарте каскадных таблиц стилей, в котором с помощью специального макроязыка один раз жестко задать форматирование страницы [10, 11].

Основным понятием CSS является стиль – т. е. набор правил оформления и форматирования, который может быть применен к различным элементам страницы. Другими словами, файл CSS выполняет роль некоего шаблона, применяемого для форматирования текста, таблиц и иных элементов в документе HTML.

Для присвоения элементу определенных характеристик необходимо один раз описать этот элемент и определить это описание как стиль. В дальнейшем следует указывать, что элемент, который необходимо оформить соответствующим образом, должен принять свойства стиля, описанного ранее. Описание стиля хранится в отдельном файле, что дает ряд преимуществ: использование описания стиля на любом количестве Web-страниц; возможность изменить оформление любого количества страниц, исправив лишь описание стиля в одном (отдельном) файле.

Кроме того, CSS позволяет работать со шрифтовым оформлением страниц на гораздо более высоком уровне, чем стандартный HTML, избегая излишнего утяжеления страниц графикой.

В [13] рассмотрен механизм, с помощью которого стили присваиваются элементам Web-страниц. Самый простой случай присвоения какому-либо элементу определенного стиля выглядит так:

*НАЗВАНИЕ\_ЭЛЕМЕНТА {свойство: значение;},*

где НАЗВАНИЕ\_ЭЛЕМЕНТА – имя HTML-тега (H1, P, TD, A и т. д.), а параметры в фигурных скобках – список свойств элемента и присвоенных им значений.

В нижеприведенном примере всем заголовкам на странице, оформленным тегом H1, присваивается размер шрифта 30 пунктов и синий цвет:

*H1 {font-size: 30pt; color: blue;}*

Элементы страниц, созданные с использованием CSS, используют механизм наследования: т. е. если располагается изображение внутри тега <P>...</P>, оформленного с помощью CSS, с отступами, так, чтобы параграф занимал только определенную часть ширины страницы, изображение также унаследует значения отступов, указанные для этого параграфа.

CSS реализует возможность присваивать стили не всем одинаковым элементам страницы, а избирательно – для этого используется параметр CLASS = "имя класса" или идентификатор ID=«имя элемента», присваивающиеся любому элементу страницы. Рассмотрим эти возможности подробнее.

Параметр CLASS применяется в случае, если необходимо создать одинаковый стиль для нескольких, но не всех элементов страницы (одинаковых или разных), например, описание стиля для класса b-c:

*.b-c {font-weight: bold; text-align: center;}*

В приведенном выше примере все элементы класса b-c будут отображаться жирным шрифтом с выравниванием по центру страницы (или ячейки таблицы).

В приведенном ниже примере параграфу присвоен стиль класса b-c:

```
<P CLASS="b-c">Текст параграфа</P>
```

Содержащийся в ячейке текст будет отображаться согласно описанию класса, т.е. ячейке таблицы присвоен стиль класса b-c:

```
<TD CLASS="b-c">текст</TD>
```

Технологию CSS можно использовать практически на любом сервере без каких-либо ограничений. Крупные недостатки у данной технологии также практически отсутствуют.

### **3.8 Технологии поиска информации в Интернет**

Перед современным пользователем сети Интернет стоит только одна проблема - какую службу или механизм поиска выбрать, принимая во внимание такие факторы, как полнота информационных ресурсов, средства поиска, обновляемость (расширяемость), дизайн.

Сегодня информационно-поисковые системы (ИПС) являются наиболее мощным механизмом поиска сетевых информационных ресурсов Интернет. Нельзя сказать, что с появлением Интернет и бурным вхождением его в практику информационного обеспечения появилось нечто принципиально новое, чего не было раньше. Теории и практике построения таких систем посвящено множество статей, основная масса которых приходится на конец 70-х - начало 80-х годов. ИПС в Интернет - это признание того, что ни иерархическая модель Gopher, ни гипертекстовая модель World Wide Web еще не решают проблему поиска информации в больших объемах разнородных документов. И на сегодняшний день нет другого способа быстрого поиска данных, кроме поиска по ключевым словам [6].

Основные причины существования ИПС в Интернет состоят в том, что объем информации в Сети в геометрической прогрессии

увеличивается; информация недостаточно или чересчур подробно структурирована; информационные массивы имеют широкий тематический профиль.

Назначение информационно-поисковых систем Интернет: *ретро-поиск* - свободный поиск информации в информационных массивах по совокупности признаков (обычно ключевых слов), *избирательное распространение информации* - поиск информации по заранее подготовленным запросам с определенной периодичностью.

Существует два вида информационных баз данных о Web-страницах: поисковые машины и каталоги:

- поисковые машины (spiders, crawlers) постоянно исследуют Сеть с целью пополнения собственной базы данных документов. Обычно это не требует никаких усилий со стороны человека. Для поисковых систем довольно важна конструкция каждого документа. Большое значение имеют title, meta-tag и содержимое страницы. Примером может быть поисковая система Altavista;

- каталоги: в отличие от поисковых машин, информация в каталоги заносится по инициативе человека. Добавляемая страница должна быть жестко привязана к принятым в каталоге категориям. Конструкция страниц значения не имеет. Примером каталога может служить Yahoo.

Документальным массивом ИПС Интернет является все множество документов шести основных типов: WWW-страницы, Gopher-файлы, документы WAIS, записи архивов FTP, новости Usenet и статьи почтовых списков рассылки. Все это довольно разнородная информация, которая представлена в виде различных, никак не согласованных друг с другом форматов данных: тексты, графическая и аудиоинформация и вообще все, что имеется в указанных хранилищах. Естественно возникает вопрос - как информационно-поисковая система должна со всем этим работать?

Первая задача, которую должна решить ИПС - это приписывание списка ключевых слов документу или информационному ресурсу. Эта

процедура называется *индексированием*. Часто, однако, индексированием называют составление файла инвертированного списка, в котором каждому термину индексирования ставится в соответствие список документов, в которых он встречается. Такая процедура является только частным случаем, а точнее, техническим аспектом создания поискового аппарата ИПС. Проблема, связанная с индексированием, заключается в том, что приписывание поискового образа документу (ПОД) или информационному ресурсу опирается на представление о словаре, из которого эти термины выбираются, как о фиксированной совокупности терминов.

В традиционных системах существовало разбиение на системы с контролируемым словарем и системы со свободным словарем. Контролируемый словарь предполагал ведение некоторой лексической базы данных, добавление терминов в которую производилось администратором системы, и все новые документы могли быть заиндексированы только теми терминами, которые были в этой базе данных. Свободный словарь пополнялся автоматически по мере появления новых документов. Однако на момент актуализации словарь также фиксировался. Актуализация предполагала полную перезагрузку базы данных. В момент этого обновления перегружались сами документы, и обновлялся словарь, а после его обновления производилась переиндексация документов. Процедура актуализации занимала достаточно много времени и доступ к системе, в момент ее актуализации, закрывался.

В Интернет, где ресурсы появляются и исчезают ежедневно, такая процедура невозможна. При создании программы Veronica для GopherSpace предполагалось, что все серверы должны быть зарегистрированы, и таким образом велся учет наличия или отсутствия ресурса. Veronica раз в месяц проверяла наличие документов Gopher и обновляла свою базу данных ПОД для документов Gopher. В WWW для

решения этой задачи используются программы сканирования сети или роботы-индексировщики. Разработка роботов - это довольно нетривиальная задача, так как существует опасность заикливания робота или его попадания на виртуальные страницы. Робот просматривает сеть, находит новые ресурсы, приписывает им термины и помещает в базу данных индекса. Главный вопрос заключается в том, что за термины приписывать документам, откуда их брать, ведь ряд ресурсов вообще не является текстом.

Сегодня роботы обычно используют для индексирования следующие источники для пополнения своих виртуальных словарей: гипертекстовые ссылки, заголовки, заглавия (H1,H2), аннотации, списки ключевых слов, полные тексты документов, а также сообщения администраторов о своих Web-страницах.

Для индексирования Telnet, Gopher, FTP, нетекстовой информации используются главным образом URL, для новостей Usenet и почтовых списков поля Subject и Keywords.

Наибольший простор для построения поискового образа документа дают HTML-документы. Однако не все термины из перечисленных элементов документов попадают в их поисковые образы. Очень активно применяются списки запрещенных слов (stop-words), которые не могут быть употреблены для индексирования, общих слов (предлоги, союзы и т.п.). Таким образом, даже то, что в OpenText, например, называется полнотекстовым индексированием, реально является результатом выбора слов из текста документа и сравнения с набором различных словарей, после которого термин попадает в ПОД, а потом и в индекс системы. Для того чтобы не увеличивать объем словарей и индексов (индекс системы Lycos уже сегодня равен 4 Тбайт), применяется такое понятие, как вес термина. Документ обычно индексируется с помощью 40 - 100 наиболее "тяжелых" терминов.

После того как ресурсы заиндексированы и система составила

массив ПОД, начинается построение поискового аппарата. Совершенно очевидно, что лобовой просмотр файла или файлов ПОД займет много времени, что абсолютно не приемлемо для интерактивной системы WWW. Для ускорения поиска строится *индекс*, которым в большинстве систем является набор связанных между собой файлов, ориентированных на быстрый поиск данных по запросу. Структура и состав индексов различных систем могут отличаться друг от друга и зависят от многих факторов: размер массива поисковых образов, информационно-поисковый язык, размещения различных компонентов системы и т.п. Например, индекс может состоять из таблицы идентификаторов страниц (*page-ID*), таблицы ключевых слов (*Keyword-ID*), таблицы модификации страниц, таблицы заголовков, таблицы гипертекстовых связей, инвертированного (*IL*) и прямого списка (*FL*). *Page-ID* отображает идентификаторы страниц в их URL, *Keyword-ID* - каждое ключевое слово в уникальный идентификатор этого слова, *таблица заголовков* - идентификатор страницы в заголовок страницы, *таблица гипертекстовых ссылок* - идентификатор страниц в гипертекстовую ссылку на эту страницу. *IL* ставит в соответствие каждому ключевому слову документа список пар "идентификатор страницы - позиция слова в странице". *Прямой список* - это массив поисковых образов страниц.

Все эти файлы, так или иначе, используются при поиске, но главным среди них является файл инвертированного списка. Результат поиска в данном файле - это объединение и/или пересечение списков идентификаторов страниц. Результирующий список, который преобразовывается в список заголовков, снабженных гипертекстовыми ссылками, возвращается пользователю в его программу просмотра Web. Для того чтобы быстро искать записи инвертированного списка, над ним надстраивается еще несколько файлов, например, файл буквенных пар с указанием записей инвертированного списка, начинающихся с этих пар. Кроме этого, применяется механизм прямого доступа к данным -

хеширование.

Для обновления индекса используется комбинация двух подходов. Первый можно назвать коррекцией индекса "на ходу" с помощью таблицы модификации страниц. Суть такого решения довольно проста: старая запись индекса ссылается на новую, которая и используется при поиске. Когда число таких ссылок становится достаточным для того, чтобы ощутить это при поиске, то происходит полное обновление индекса - его перезагрузка. Эффективность поиска в каждой конкретной ИПС определяется исключительно архитектурой индекса, что как правило, является "секретом фирмы".

Индекс - это только часть поискового аппарата, скрытая от пользователя. Второй частью этого аппарата является информационно-поисковый язык. Наиболее развитым языком запросов из современных ИПС Internet обладает AltaVista. Кроме обычного набора AND, OR, NOT эта система позволяет использовать еще и NEAR, позволяющий организовать контекстный поиск. Все документ в системе разбиты на поля, поэтому в запросе можно указать, в какой части документа пользователь надеется увидеть ключевое слово: ссылка, заглавие, аннотация и т.п. Можно также задавать поле ранжирования выдачи и критерий близости документов запросу.

Важным фактором является способ представления информации в программе-интерфейсе. Различают два типа интерфейсных страниц: страницы запросов и страницы результатов поиска. При составлении запроса к системе используют либо меню - ориентированный подход, либо командную строку. Первый позволяет ввести список терминов, обычно разделяемых пробелом, и выбрать тип логической связи между ними. Логическая связь распространяется на все термины. Из рис. 6.1 видно, что имеется возможность сохранить запросы пользователя. В большинстве систем это просто фраза на ИПЯ, которую можно расширить за счет добавления новых терминов и логических операторов. Но это только один

способ использования сохраненных запросов, называемый расширением или уточнением запроса. Для выполнения этой операции традиционная ИПС хранит не запрос как таковой, а результат поиска - список идентификаторов документов, который объединяется/пересекается со списком, полученным при поиске документов по новым терминам. К сожалению, сохранение списка идентификаторов найденных документов в WWW не практикуется. Это было вызвано особенностью протоколов взаимодействия программы-клиента и сервера, не поддерживающих сеансовый режим работы.

Результат поиска в базе данных ИПС - это список указателей на удовлетворяющие запросу документы. Различные системы представляют этот список по-разному. В некоторых выдается только список ссылок, а в таких, как Lycos, AltaVista и Yahoo, дается еще и краткое описание, которое заимствуется либо из заголовков, либо из тела самого документа. Кроме этого, в некоторых случаях система сообщает, насколько найденный документ соответствует запросу, например, в Yahoo - это количество терминов запроса, содержащихся в ПОД, в соответствии с которым ранжируется результат поиска. Система Lycos выдает меру соответствия документа запросу, по которой производится ранжирование.

При обзоре интерфейсов и средств поиска нельзя пройти мимо процедуры коррекции запросов по релевантности. *Релевантность* - это мера соответствия найденного системой документа потребности пользователя. Различают формальную релевантность и реальную. Первую вычисляет система, и на основании чего ранжируется выборка найденных документов. Вторая - это оценка самим пользователем найденных документов. Опыт показывает, что релевантные документы не всегда полностью соответствуют информационной потребности пользователя, что приводит к процедуре коррекции запросов (в разных системах это делается поразному).

Кроме ссылок на документы в списке, полученном пользователем, могут оказаться ссылки на части документов или на их поля. Это происходит при наличии ссылок типа `http://host/path#mark` или ссылок по схеме WAIS. Возможны ссылки и на скрипты, но обычно такие ссылки роботы пропускают, и система их не индексирует.

Если с `http`-ссылками все более или менее понятно, то ссылки WAIS - это гораздо более сложные объекты. Дело в том, что WAIS реализует архитектуру распределенной информационно-поисковой системы, при которой одна ИПС, например Lycos, строит поисковый аппарат над поисковым аппаратом другой системы - WAIS. При этом серверы WAIS имеют свои собственные локальные базы данных. При загрузке документов в WAIS администратор может описать структуру документов, разбив их на поля, и хранить документы в виде одного файла. Индекс WAIS будет ссылаться на отдельные документы и их поля как на самостоятельные единицы хранения, программа просмотра ресурсов Интернет в этом случае должна уметь работать с протоколом WAIS, чтобы получить доступ к этим документам.

В различных публикациях, посвященных конкретным системам, приводятся схемы ИПС, которые отличаются друг от друга только способом применения конкретных программных решений, а не принципом организации различных компонентов системы [ 6, 11].

Одной из причин, по которой развитие системы WWW базируется на архитектуре ИПС, является то, что поиск строится на основе преобразования предложений некоторого информационно-поискового языка (ИПС) в запросы к информационной системе.

Выделяют следующие типы информационно поисковых языков: *традиционные; взвешивание терминов; ИПЯ типа "Like this"*. Традиционное определение процедуры поиска документов в ИПС было введено для решения проблемы автоматического индексирования документов. Например, запрос к системе "Найти все документы, в которых

встречается термин "вычислительные" и "системы", либо термин "BC", но не встречается термин "комплексы"" выглядит следующим образом:

*((вычислительные and системы) or BC) not комплексы*

Такая форма запроса имеет недостатки: плохая масштабируемость выдачи, OR приводит к слишком большому расширению списка релевантных документов, а AND резко сужает отклик. Поэтому существуют модификации традиционного языка - взвешенный запрос, когда каждому термину приписывают некоторый вес.

Суть его в том, что для индексирования используют те термины, которые имеют высокую частоту встречаемости внутри документа и низкую во всем информационном массиве. Сама характеристика вычисляется как отношение частоты встречаемости термина в документе к частоте встречаемости термина в массиве. Используя эту меру, системы индексирования приписывают документу первые 20-40 символов, которые и составляют его поисковый образ. Экспериментальные результаты, позволяют сделать два вывода:

- для того чтобы использовать взвешивание следует иметь насыщенный словарь;
- термины индексирования находятся в окрестностях максимума частотного распределения терминов.

Насыщение словаря - очень важное свойство систем со свободным словарем. Говорить вообще о векторной модели информационного потока и ее применимости для информационных систем можно только в том случае, когда мощность словаря (число представленных в нем терминов) фиксирована. В локальных информационных системах вопрос о размере словаря не стоял, т.к. с момента загрузки документов и до момента актуализации информационный массив и словарь системы не менялись, и, следовательно, были фиксированными. В Интернет нет единого информационного массива, который можно было бы сразу загрузить, построив долгоживущий индекс. Поэтому система постоянно

осуществляет сканирование сети и коррекцию своего поискового аппарата - словаря, который определяется индексом и постоянно изменяется. Из-за отсутствия единой информационной службы нельзя организовать систему с контролируемым словарем.

Существуют и подавляющем большинстве применяются информационно-поисковые языки типа "Like This". Данный подход хорошо известен как вычисление мер близости "документ-запрос". В современных распределенных ИПС Интернет реально используются только 6 из 24 мер близости. Наиболее часто применяется косинус угла между поисковым образом документа и запросом пользователя. Обычно эти проценты соответствия документов запросу и выдаются в качестве справочной информации со списком найденных документов.

Начало применению запросов типа "Like This" положила система WAIS. Именно в ней был впервые сформулирован отказ от использования традиционных информационно-поисковых языков булевого типа, и было заявлено о переносе центра тяжести информационного поиска на языки, основанные на вычислении меры близости "документ-запрос". Основная цель такого подхода - желание снять с пользователей заботу по формулированию запросов на информационно-поисковых языках и дать им возможность использовать обычный естественный язык.

Для коррекции результатов поиска используются следующие способы: фильтрация; коррекция по релевантности; кластеризация. Важным способом улучшения качества поиска в информационно-поисковых системах Интернет стала процедура коррекции запроса по релевантности. Пользователю предоставлялась возможность отметить документы, которые являлись релевантными его запросу. После этого запрос расширялся терминами этих документов и снова вычислялось значение для поисковых образов документов всего массива [6]. В рамках линейной модели индексирования и поиска эта процедура может быть также выражена через матричные выражения (Lycos, OpenText, и Altavista,

и другие системы Интернет применяют линейную модель индексирования и поиска).

В литературе по информационному поиску часто можно встретить термин "профиль", который относят к запросам пользователей. Информационный профиль или тематический профиль имеется и у информационной системы. Наиболее просто тематический профиль системы материализуется в виде классификации, которая применяется в данной системе или рубрикаторе, в некоторых ИПС профиль играет еще и роль навигационного средства, позволяющего получить доступ непосредственно к набору документов, попадающих в тот или иной раздел классификации. Многие системы Интернет имеют несколько профилей, которые могут быть соотнесены с фасетной классификацией.

Отличия в стратегии и широте охвата материала поисковых средств часто приводят к тому, что разные системы дают разные ответы на один и тот же запрос. Этим не замедлили воспользоваться разработчики поисковых средств особого рода - *мета-машин*, основанных исключительно на применении потенциала других поисковых систем (например, Cyber 411, Highway 61, MetaCrawler).

Наряду с развитием универсальных поисковых средств, начинают создаваться средства узкой отраслевой направленности.

### **3.9 Информационные порталы**

Рождение информационного портала (ИП) как самостоятельной информационной единицы связано с появлением нового класса коммерческих сайтов - информационных накопителей. Первое упоминание слова портал связано с выпуском компанией Merlin Lynch отчёта, в котором впервые было введено понятие корпоративного информационного портала – EIP (Enterprise Information Portal), под которым понимался пакет программного обеспечения, дающий компании возможность обеспечивать доступ к ресурсам внешних и внутренних сетей.

Серьёзная потребность в построении ИП возникла при переходе корпоративных БД к сетевом формату. Поэтому возникла потребность систематизировать и усовершенствовать информационные хранилища.

В настоящее время в определении термина портал ведущие специалисты в области ИТ не пришли к единому мнению. Приведем несколько определений.

«Портал – это сайт, организованный как системное многоуровневое объединение различных ресурсов и сервисов» (информационно-консалтинговый центр по электронной коммерции e-Commerce).

«Портал — централизованная платформа для постоянного доступа к информации и всем необходимым приложениям» (рекламное агентство Dot).

Интернет-издание «Сетевой montly» считает, что: «Портал – это Web-сайт, обеспечивающий:

- объединение информационного наполнения и доставку информации;
- совместную работу и коллективные услуги;
- доступ к услугам и приложениям на основе строгой персонализации.

Информацию в портале обычно организуют по иерархическому признаку, связанному с определенной тематикой. Различают:

- вертикальные порталы (специализированные);
- горизонтальные порталы (универсальные);
- потребительские порталы;
- торговые порталы;
- корпоративные порталы.

Вертикальные порталы отличаются узкой тематической направленностью, посвященные определенной тематике.

Информация горизонтальных порталов носит общий характер, в них обозревают различные темы.

Потребительские порталы содержат большое количество ссылок на различные сайты, имеют широкие поисковые возможности и служат для более удобного пребывания пользователя в Сети. Некоторые из них предоставляют услуги электронной почты, сервисы для планирования времени, ведения баз данных адресов и дат, а также другие бесплатные приложения. В качестве примера можно привести Yahoo, Rambler, InfoArt.

Основной задачей торговых порталов является объединение вокруг себя наибольшего количества узлов поставщиков и покупателей и централизация документооборота между ними. Все электронные транзакции в таких порталах проходят через электронные торговые площадки, соединяющие покупателей и поставщиков по «звездообразной топологии».

Корпоративные порталы предназначены для организации единого информационного пространства внутри компании.

Привлекательность информационного портала состоит в том, что они позволяют максимально приблизить бизнес к клиентам, партнёрам и поставщикам различных услуг, обеспечить пользователям персонализацию и возможность устанавливать отношения внутри рабочих и информационных групп.

### **1) Структура портала**

Принятие решения о создании конкретного портала должно основываться на понимании того, кто составит его целевую аудиторию, какую информацию посетители найдут на его страницах, кому будет поручена ее актуализация, какие услуги смогут получить граждане с помощью этого портала.

В зависимости от требуемой сложности сервисов и масштаба решения, Интернет-портал может включать компоненты различного назначения. Но во всех случаях он размещается на компьютере или даже кластере компьютеров, имеющих постоянное скоростное соединение с Интернетом.

Основу портала составляют сервлеты Java. Сервлеты используют HTTP для соединения с браузерами пользователей. Вывод содержимого портала может быть потоком XML. XSL форматирует поток XML для корректного отображения содержимого портала на специфичных WEB устройствах, отличных от стандартных web-браузеров, например КПК или в WAP - браузерах сотовых телефонов.

Все содержимое портала можно разделить на службы переднего плана - общедоступный web-сервер с диалогом авторизации пользователей и службы заднего плана - корпоративные приложения, базы данных, файлы и другой информации, доступные пользователям только после авторизации в портале. Например, для авторизации портал Novell Portal Services использует каталог eDirectory или любой соответствующий стандарту Lightweight Directory Access Protocol версии 3 (LDAPv3). Для создания информационного наполнения «контента» web - страниц портал использует информацию о правах доступа пользователей к программам, файлам, документам, хранящихся в каталоге. В зависимости от того, является ли пользователь сотрудником, клиентом или поставщиком, этот пользователь получит доступ только к той части информации портала, права на которую ему определены в каталоге. Кроме того, портал хранит в каталоге информацию о представлении порталных страниц. В результате пользователи сами могут настраивать внешний вид портала, используя набор predeterminedных тем. Под термином «тема» понимается набор файлов с описанием таблиц стилей (.xsl), скриптов, фоновых рисунков, картинок, которые используются для создания внешнего вида документа XML. Связь между информационным наполнением портала и различными сетевыми сервисами, такими как доступ к корпоративным приложениям, базам данным, файлам, почте реализуют gadget - компоненты. Gadgets компоненты представлены в каталоге объектами, которые хранят указатели на код компонентов портала – сервлеты (Java программа (несамостоятельная), которая обращается к различным источникам данных

и выводит эти данные в форме потока XML). Такими данными могут быть файловая система, приложения, какие – либо WEB – сервисы.

Для работы портала необходима серверная операционная система, обеспечивающая надежность работы, возможность параллельного исполнения множества задач, поддержку транзакционности.

Для работы Интернет-портала необходим также веб-сервер. В относительно простых случаях это может быть просто программный сервер для доставки пользователям страниц по Интернет-протоколу HTTP. Если требуется высокая адаптируемость к пиковым нагрузкам, возможность персонализации демонстрируемых страниц, обеспечение сложных деловых операций (например, тендеров), то лучше выбрать специальный сервер приложений со встроенными элементами бизнес-логики.

Большинство документов портала хранится в базе под управлением СУБД (системы управления базами данных). Это обеспечивает удобство хранения и модификации, а также поиска и представления материалов.

Кроме того, необходима та или иная система управления контентом (информационные ресурсы, информационное наполнение web-узла). Она позволяет служащим, не являющимся специалистами в информационных технологиях, публиковать документы быстро и качественно, а также поддерживать их актуальность. Обычно содержание («контент») и способ представления («шаблоны») документов хранятся отдельно и связываются с помощью описаний, основанных на стандарте XML. Это позволяет при необходимости выводить информацию не только на персональные компьютеры, но и, например, на сотовые телефоны.

В случае если через портал идет большой объем запросов, требующих обработки несколькими служащими или отделами и контроля исполнения, или если портал представляет в сети Интернет несколько государственных учреждений, необходимо использование сервера интеграции приложений или, как его еще называют, коммуникационного

«моста» («шлюза») между несколькими информационными источниками. Сервер интеграции обычно поддерживает преобразование документов через промежуточные XML-форматы.

Конкретные требования к этим программным модулям определяются точным перечнем задач, которые должен решать Интернет-портал.

Средства развёртывания, которые используются для портала, - выбираются в зависимости от масштаба решаемой задачи и масштаба предприятия.

## **2) Организация порталов**

Если размещаемые на портале документы существенно влияют на условия хозяйствования и/или права и обязанности граждан и создаются в достаточно сложных и тщательно регламентированных процессах делопроизводства (как обычно и бывает), то целесообразно готовить их к публикации не в отдельной системе контент-менеджмента, а в предназначенной специально для этого системе делопроизводства.

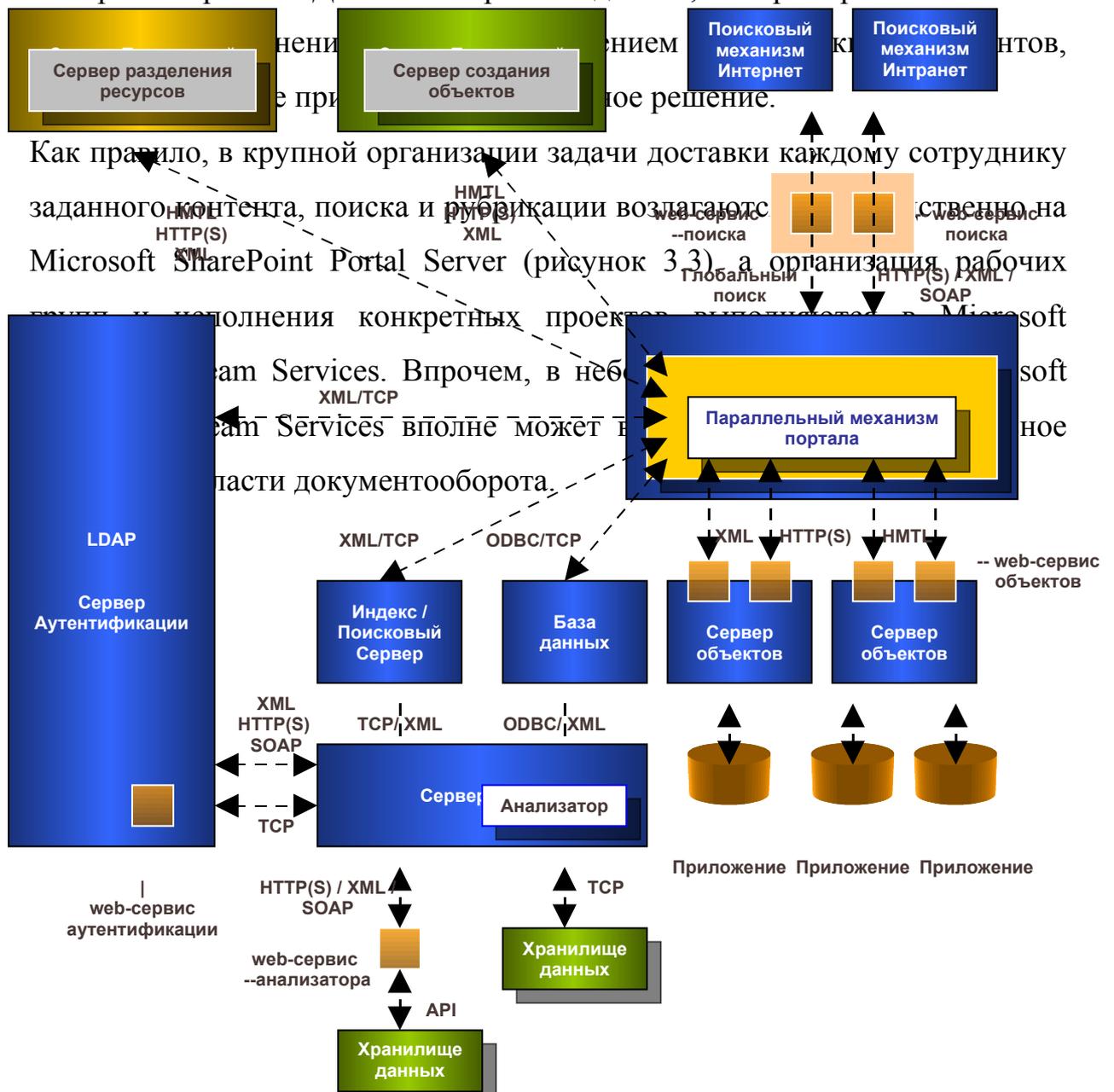
Интеграция Интернет-портала с современной системой документооборота интранет-портала позволяет ускорить и упростить весь цикл разработки нормативных документов — от выдачи поручения до публикации. Это решение незаменимо для учреждений, готовящих и/или распространяющих большой объем нормативных актов

Современные внутренние системы документооборота объединяют черты традиционных офисных систем (разработка документов ведется в привычных офисных приложениях) и порталов (настраиваемый пользовательский интерфейс, собираемый из компонентов; поиск по всем типам и местам расположения документов; возможность автоматической рубрикации материалов). Фактически, современная система документооборота это — интранет-портал, интегрированный с офисными приложениями. При этом от большинства работников многие мощные средства (в частности, организации маршрутов делопроизводства) могут быть скрыты за простым и очевидным интерфейсом. Например,

помещение подготовленного документа в некоторую «продвинутую» папку «На согласование» может автоматически вызывать процесс направления документа уполномоченным для визирования сотрудником, рассылки им уведомлений, контроля срока исполнения и пр.

В такой конфигурации в систему управления контентом попадают уже практически готовые документы из интранет-портала документооборота. Остается только определить им место размещения на портале, способ форматирования, снабдить уместными ссылками и, наконец, «дать путевку в жизнь», формально разрешив публикацию.

Именно объединение интранет-портала и системы управления контентом Интернет-портала позволяет с минимальными затратами и ошибками наиболее оперативно публиковать множество документов на Интернет-портале. Для некоторых ведомств, например связанных с



### Рисунок 3.3 – Общая структура корпоративного портала

Корпоративный портал SharePoint базируется на архитектуре web-сервисов. Использование этих стандартов позволяет разработчикам создавать масштабируемую, открытую и расширяемую систему.

Web-сервис - это программа, использующая Интернет-протоколы для коммуникации с другими компонентами или программами. Их задача - предоставлять пользователям доступ к различным приложениям и данным (Word, Excel, Access, Lotus Notes, ERP, CRM-системы, электронная почта, специализированные приложения и т.д.).

Ядро корпоративного портала SharePoint состоит из следующих компонентов:

-SharePoint Portal Application - предназначен для формирования отображения страниц пользователя, рабочих групп и директории документов, представления сервисов портала и обработки запросов пользователей.

-Parallel Portal Engine (Механизм параллельной обработки) - позволяет одновременно получать доступ к информационным ресурсам и сервисам, расположенным на большом количестве различных web-серверов. Благодаря механизму параллельной обработки достигается высокая производительность и высокая надежность работы. Испытание на масштабируемость, проведенное Microsoft, показало, что корпоративный портал SharePoint способен одновременно поддерживать сотни тысяч пользователей и ежедневно обрабатывать свыше 30 миллионов обращений к portalу.

-SharePoint Database (База данных корпоративного портала SharePoint) – представляет из себя хранилище данных, составляющих основу портала. Здесь содержится информация о настройках пользователей и правилах доступа к ресурсам, всех объектах портала SharePoint , а также структуре директории «Документы». База данных корпоративного портала SharePoint не хранит документы, доступные посредством портала. Исходные документы остаются на своих изначальных местах или загружаются в один из серверов общего доступа SharePoint .

-Gadget Server (Сервер модулей Gadget) – позволяет с помощью специальных модулей (гэджетов или вэб-частей), представляющих собой информационные модули, которые встраиваются в портал, связываться с приложением или каким-либо Интернет-сервисом и выводить на экран данные из этих приложений и сервисов. В поставку корпоративного портала SharePoint входит более тысячи гэджетов для наиболее часто используемых программ (Lotus Domino, MS Exchange, MS Excel, SAP, Siebel и т.д.). Кроме того, существуют программные пакеты стандартных гэджетов для большинства корпоративных приложений. В случае необходимости можно самостоятельно разработать новый гэджет с помощью встроенных средств, входящих в стандартную поставку.

-SharePoint Studio Server (Студио Сервер) - представляет собой среду разработки приложений, используемую для создания и управления

гэджетами. Гэджеты можно создавать с помощью шаблонов и с использованием «Помощника».

-SharePoint Collaboration Server (Сервер совместной работы) - дает возможность пользователям совместно работать над проектами: составлять графики выполнения работ, работать над документами, обмениваться идеями. Сервер может объединять информацию об активных проектах со всеми ресурсами, интегрированными в портал, включая электронную почту, отчеты о продажах, телефонные списки и информацию о конкурентах. В результате мы получаем среду, которая может использоваться сотрудниками кафедры, ее партнерами и студентами в качестве рабочей платформы для проектных групп и подразделений. Сервер совместной работы обладает значительным диапазоном сервисов, предназначенных для организации совместной работы на портале, в виде гэджетов. Руководитель проекта, например, может выбирать гэджеты, необходимые для совместной работы над каким-либо проектом, объединить их на отдельной странице портала.

-SharePoint Content Server (Контент-сервер) - представляет собой ведущую систему управления контентом (содержимым) портала, включая внутренние и внешние корпоративные информационные системы и интернет-сайты, обеспечивая безопасность, возможности навигации и функции системы управления знаниями. Он собирает информацию от пользователей и пересылает ее для утверждения. Затем эта информация в виде web-страниц публикуется в директории «Документы» или на внутренних и внешних корпоративных сайтах, доступных на портале. Кроме того, Контент-сервер может публиковать информацию в различных областях внутри портала. Контент-сервер обеспечивает защиту каждого объекта в системе - от информационных данных и ссылок до таких сервисов, как задания и шаблоны. Это позволяет пользователям просматривать только ту информацию и получать те возможности, которые определяются их полномочиями.

-SharePoint Search Server (Сервер поиска) - индексирует данные, включая документы, хранимые на Сервере совместной работы, web-страницы, управляемые Контент-сервером, и другие проиндексированные данные из файловых систем, web-сайтов и баз документов, находящихся в директории «Документы». Может запускаться с любого компьютера, у которого есть доступ к файлам общего доступа Корпоративного портала SharePoint. Поисковый механизм полностью русифицирован, что позволяет осуществлять поиск с поддержкой морфологии русского языка.

### **3) Организация простого портала**

Простой портал, как правило, это портал органа местной власти. Часто местные органы не имеют достаточно специалистов и средств, чтобы сразу построить современный полнофункциональный портал. Однако это не основание, чтобы сразу отказаться от всей информатизации вообще. Следует внедрить простое, но «правильное» решение начального уровня.

В данном случае «правильное» решение означает такое, которое позволяет удовлетворить не только уже имеющиеся потребности граждан, но и те, которые могут появиться в будущем. Для этого важно сразу выбрать оптимальную программную платформу, которая допускает постепенное наращивание и развитие без необходимости переделки «с нуля», которая уже использовалась в аналогичных проектах и для работы с которой имеется достаточное количество специалистов.

Простой портал территориального органа управления включает web-сервер и систему управления базой данных. Это, конечно, не совсем «настоящий» портал. На нем трудно публиковать значительный объем информации.

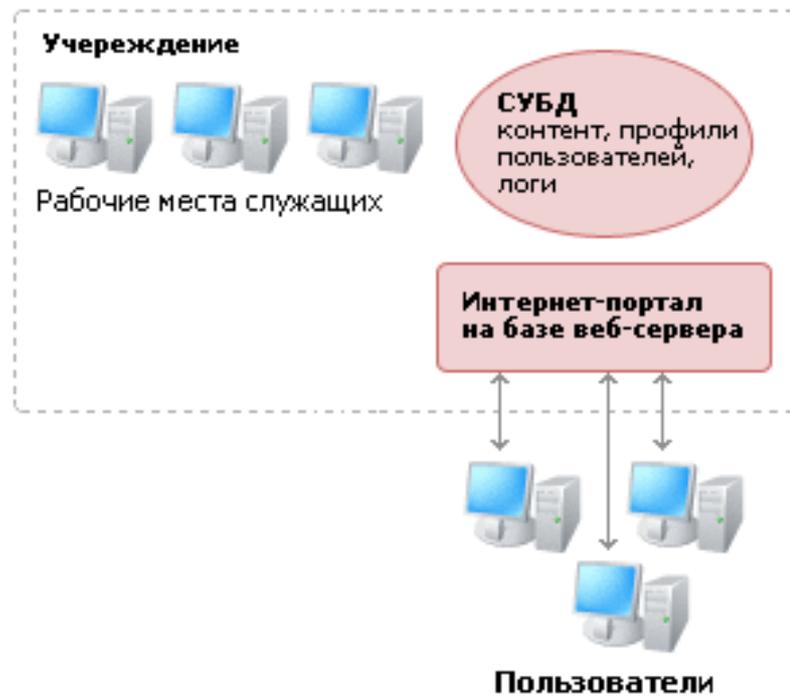


Рисунок 3.4- Структура простого портала

Наиболее широко распространенная платформа для подобных решений — Microsoft Windows 2000 Server. По мере роста портала к ней можно добавлять необходимые дополнительные серверы семейства .NET Enterprise Servers. Как правило, любому portalу сразу же необходима СУБД Microsoft SQL Server 2000. Для публикации материалов на несложном муниципальном портале можно использовать Microsoft Word XP, FrontPage или простейшую специализированную систему управления контентом, с использованием технологии ASP6.

Даже самое простое решение уже позволяет добиться нескольких важных результатов:

- улучшить отношение граждан к учреждению;
- привлечь или закрепить на государственной службе лучшие кадры.

Следует признать, что это - еще не совсем портал. Такое решение применимо только в тех случаях, пока учреждение не готово предоставить гражданам действительно большой объем динамически обновляемой

информации и в режиме реального времени принимать от них запросы или заполненные формы.

#### **4) Портал, ориентированный на обработку сложных запросов**

Такой портал обычно создается в министерстве или его территориальном отделении для онлайн-обработки сложных запросов или документарных форм (например, налоговых деклараций), поступающих от граждан или организаций. Его отличительные особенности:

- необходимость использования дополнительных программных приложений (возможно, специфичных только для данного ведомства, особо сложных или даже устаревших, но слишком дорогостоящих в модернизации);

- частое изменение процессов делопроизводства в связи с изменениями законодательной базы и подзаконных актов, регулирующих обработку запросов или документарных форм;

- развитый документооборот;

- необходимость отслеживать исполнение связанных групп операций и в случае возникновения сбоя или ошибки при выполнении одной из операций отменять все предыдущие шаги, связанные в транзакцию. Наиболее типичной областью применения транзакций являются финансовые операции;

- жесткий контроль исполнения.

На рисунке 3.5 приведена структура портала, ориентированного на обработку сложных запросов.

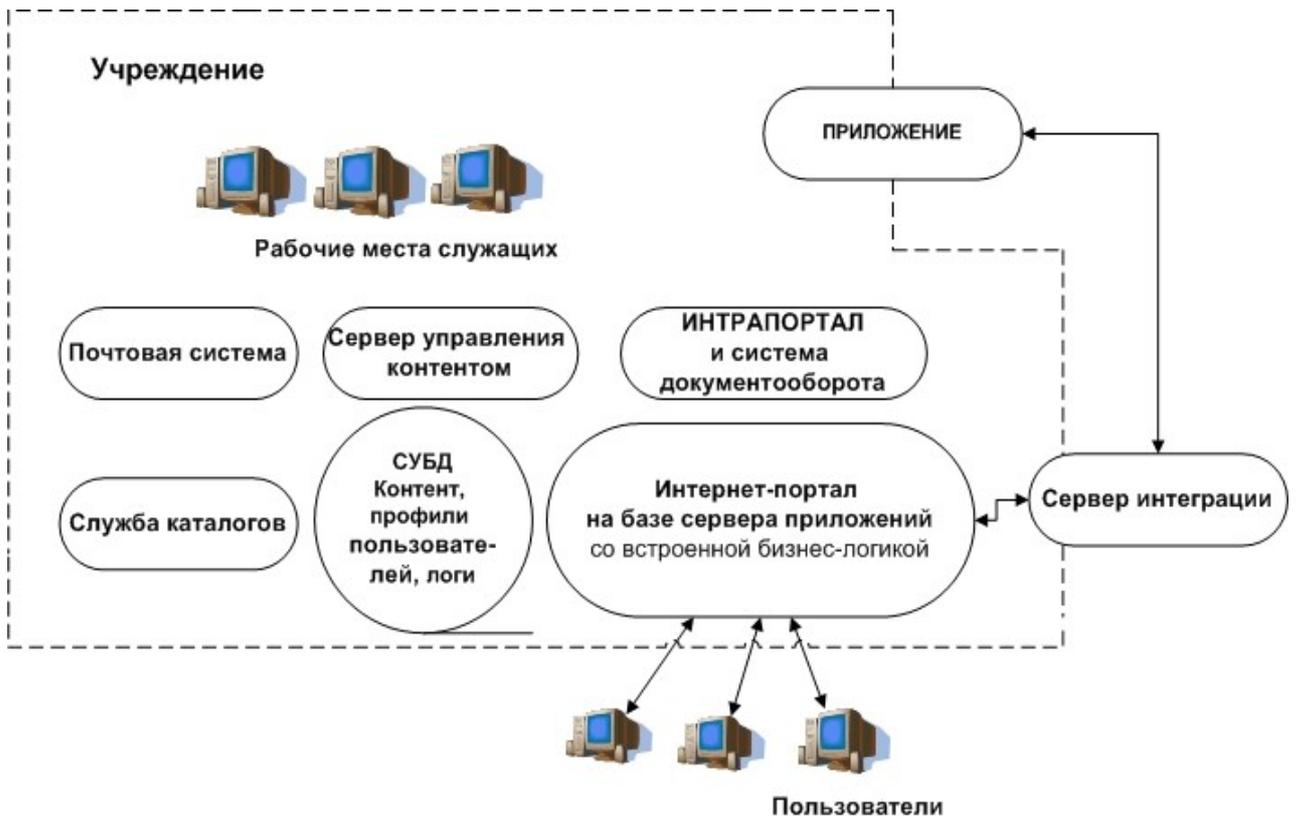


Рисунок 3.5 – Структура портала с обработкой сложных запросов

Использование сервера интеграции позволяет относительно простым и регулярным образом объединять все элементы информационной инфраструктуры учреждения, включая Интернет- и интранет-порталы, а также любые специализированные приложения. Сервер интеграции необходим для организации обработки потоков запросов и документарных форм от граждан и организаций. Он также значительно сокращает время переналадки информационной системы учреждения в процессе его структурной реорганизации.

Чтобы портал оказал серьезную помощь в работе такой организации, он должен обеспечить:

- прием информации от граждан и предприятий. В некоторых случаях такие обращения могут потребовать электронной цифровой подписи или иных средств идентификации, например смарт-карт. Данные вопросы регулируются российским законодательством;

-обработку информации в рамках внутренних процессов делопроизводства и документооборота;

- выдачу гражданам соответствующих результатов обработки, что, как правило, происходит как в электронной, так и в бумажной форме (например, Патентное ведомство может принимать заявки на изобретения через портал, вести переговоры с изобретателями и патентными поверенными через портал, уведомлять их о решении через портал, но отсылать патент заказным почтовым отправлением).

Данная категория порталов требует определенных мер обеспечения безопасности информации, для чего рекомендуется использовать современное, профессионально разработанное, программное обеспечение с постоянно совершенствуемой и обновляемой защитой для ликвидации проблем уязвимости.

Для учета всех этих особенностей и решения вытекающих задач построения гибкой и надежной архитектуры портала с маршрутизацией и обработкой сложных запросов служит специальный класс серверного программного обеспечения — серверы интеграции (называемые также иногда «мостами» или «шлюзами»). До их появления задача интеграции множества разнородных приложений в крупном учреждении могла затянуться на несколько лет. Сейчас она решается за несколько месяцев.

Сервер интеграции можно использовать и для объединения Интернет-портала с интранет-порталом. Это позволяет добиться большей регулярности и гибкости решения, особенно в учреждениях, периодически претерпевающих реорганизацию.

Сервер интеграции принимает входящие документы, конвертирует их в подходящий формат (правила преобразования задаются с помощью XML-схем), передает соответствующим приемникам (в качестве которых могут выступать как серверные приложения, так и офисные программы, с которыми работают пользователи), следит за своевременной обработкой в соответствии с заданными сценариями и маршрутами делопроизводства.

Сервер интеграции обычно снабжается визуальными инструментами, позволяющими буквально «нарисовать» в наглядном и понятном как ИТ-специалистам, так и руководителям организации виде все процессы документооборота в учреждении.

Для портала, создаваемого на базе технологий Microsoft, наиболее естественным решением является использование сервера интеграции Microsoft BizTalk 2000. Он позволяет маршрутизировать документы, контролировать транзакционность и гарантировать доставку адресату (путем проверки условий доставки и повторных попыток, а также уведомлений). Кроме того, он снабжен наглядными средствами преобразования документов из одной формы в другую, а также проектирования бизнес-процессов.

Портал с сервером интеграции позволяет решать очень сложные и ответственные задачи, включая решение вопросов пособий и льгот, распределения ресурсов, приема деклараций и отчетов и пр. [11,13].

### **Контрольные вопросы**

- 1) Какие технологии относятся к Web- технологиям?
- 2) Перечислите назначение и отличия Web-технологий.
- 3) Какие технологии используются при сложном форматировании html- документа? Поясните их отличия.
- 4) Перечислите методы реализации технология CGI.
- 5) Какие возможности предоставляет технология PHP?
- 6) Поясните способы отделения PHP-кода от общего кода HTML.
- 7) Сформулируйте правила для выбора имени переменной в PHP.
- 8) Поясните способы ввода данных пользователем.
- 9) Как выполняется проверка вводимой пользователем информации?
- 10) Что понимается под технологией Active-X?
- 11) Перечислите основные компоненты технологией Active-X.
- 12) Что понимается под термином «информационный профиль»?

- 13) Перечислите основные задачи ИПС.
- 14) Что является результатом поиска в базе данных ИПС?
- 15) Что понимается под индексированием ресурса?
- 16) Поясните термин информационный портал.
- 17) Приведите классификацию информационных порталов.
- 18) Что лежит в основе портала?
- 19) Поясните влияние сложности сервисов и масштаба решения на выбор структуры портала.
- 20) Каково назначение сервера интеграции в портале?
- 21) Перечислите основные компоненты ядро корпоративного портала на базе SharePoint.
- 22) Назначение контент-сервера корпоративного портала.

## **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

- 1) Джамса К., Коуп К. Программирование для Internet в среде Windows.- СПб.: Питер, 1996.-688 с.
- 2) Котеров Д.В. Самоучитель РНР 4.- СПб.: БХВ-Петербург, 2001. - 576 с.

- 3) Левин В.И. Основы информационных технологий: Интернет-университет информационных технологий, ИНТУИТ.ру , БИНОМ. Лаборатория знаний, 2007. – 336 с.
- 4) Мазуркевич А., Еловой Д. PHP: настольная книга программиста - Мн.: Новое знание, 2003. - 480 с.
- 5) Романчева Н.И., Соломенцев В.В. Информационные ресурсы: Интернет. Учебное пособие – М.: РИО МГТУ ГА, 2000, 118с.
- 6) Дж. Солтон. Динамические библиотечно-информационные системы. - М.: Мир, 1979
- 7) Семенов Ю.А htmПротокол Интернет для работы с сообщениями IMAP , <http://book.itep.ru/4/44/imap4443>.
- 8) Храпцов П.Б. , Брик С.А. , Русак А.М. , Сурин А.И. Основы web-технологий .- Интернет-университет информационных технологий - ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007
- 9) Sollins, K.R. TFTP Protocol (revision 2), RFC-783, MIT, June 1981
- 10) <http://www.djinn.ru/forum/viewforum.php> -История информационных технологий.
- 11) <http://www.intuit.ru/department/network/pdsi/lit.html> - Интернет университет
- 12) <http://microsoft.com/php>
- 13) <http://www.diwaхх.ru/web/osnoviweb3.php> - Интернет-портал Diwaхх
- 14) <http://www.netservice.ru> - Управление интернет проектами.

ПРИЛОЖЕНИЕ  
Контрольные тесты

1) Укажите номер последовательности, содержащей неправильную запись IP адреса по версии v6:

- |                           |                               |
|---------------------------|-------------------------------|
| а) ::FFFF:129.144.52.38   | б) FFDC:A98::765:321          |
| в) FFDC:0A98::765:321     | г) 0:0:0:FFFF:0:192.144.52.38 |
| д) ::FFFF:0:192.144.52.38 |                               |

2) Укажите номер строки, содержащей неправильное использование команды telnet:

- |                                |                                |
|--------------------------------|--------------------------------|
| а) tn 3270 microsoft.com       | б) telnet microsoft.com        |
| в) telnet tigger.ivnc.net 1840 | г) telnet madlab.sprl.edu 3000 |
| д) telnet 25                   |                                |

3) Укажите номер неправильного ответа.

- а) Почтовые сообщения доставляются клиенту по протоколу POP, а посылаются при помощи протокола SMTP.
- б) На компьютере пользователя существует два отдельных клиента-интерфейса к почтовой системе: SMTP и POP.
- в) На компьютере пользователя существуют отдельные клиенты (Split UA).
- г) После того, как почтовая программа установила TCP-соединение с портом протокола POP3 (110) начинается стадия транзакции.
- д) Во главе модели POP3 находится отдельный персональный компьютер, работающий исключительно в качестве клиента почтовой системы.

4) Укажите время, когда электронное письмо, помещенное в 2.00 на SMTP-сервере в пункте А сможет дойти до пункта С, при наличии в пункте В промежуточного почтового сервера. (Следует учесть, что связь между участками Internet в пункте А и пункте В плохая с 10.00 до 18.00, а между В и С - с 2.00 до 10.00)?

- |                  |                  |
|------------------|------------------|
| а) в 18.00 часов | б) в 10.00 часов |
| в) в 2.00 часов  | г) в 00. часов   |

5) Командами контроля доступа протокола FTP являются:

- а) USER, PASS, ACCT, CWD, CDUP, SMNT, REIN.
- б) REST, PASV, STOR, MODE, LIST, PORT, TYPE, ALLO, RETR.
- в) USER, PASS, ACCT, REST, LIST, RETR.
- г) USER, PASS, ACCT, CDUP, LIST, SMNT.

д) CWD, USER, PASS, TYPE, STRY, MODE .

6) Укажите номер правильного ответа для нижеприведенных утверждений.

1 - Telnet позволяет потоку команд идти в обоих направлениях (клиент-сервер, сервер-клиент).

2- Клиенты SMTP, POP3, Telnet посылают строки символов NVT ASCII, которые являются командами протокола и ждут ответа от сервера.

3 - В любой момент времени любая сторона Telnet- соединения может передавать Telnet-команды.

4 - Telnet использует строки символов NVT ASCII для представления команд.

5 - Telnet передает команды как специально определенные управляющие эскап-последовательности.

- а) утверждения 1,3,5 - верные, 2,4 - ложные
- б) утверждения 1,2, 4 - верные, 3,5 - ложные
- в) утверждения 1,2 - верные, 3,4,5 - ложные
- г) утверждения 3,4,5- верные, 1,2 - ложные
- д) утверждения 2,3,4 - верные, 1,5 - ложные
- е) утверждения 3,4 - верные, 1,2,5 – ложные

7) Укажите номер неправильного ответа.

Адрес типа anycast (IPv6):

- а) может быть назначен только интерфейсам маршрутизатора
- б) ориентирован на применение маршрутизации от источника (Source Routing)
- в) пределяет группу интерфейсов
- г) определяет уникальный идентификатор отдельного интерфейса конечного узла или маршрутизатора
- д) позволяет передавать пакет через ближайший маршрутизатор провайдера

8) Обратная запись элемента 4321:0:1:2:3:4:567:89AB (для Ipv6) в дереве домена соответствует:

- а) 4.3.2.1.0.1.2.3.4.5.6.7.8.9.A.B.IP6.INT
- б) V.A.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.1.2.3.4.IP6.INT
- в) IP6.INT. 4.3.2.1.0.1.2.3.4.5.6.7.8.9.A.B.
- г) VA98:765:4:3:2:1:0:123.4:IP6.INT
- д) VA98.765.4.3.2.1.0.123.4.IP6.INT
- е) V.A.9.8.7.6.5.0.4.3.2.1.0.1.2.3.4.IP6.INT

9) Укажите номер неверного ответа:

а) Поле префикса формата (FP) для типа unicast адресов Ipv6 состоит из 3-х бит.

б) Структура агрегированных адресов Ipv6 включает 6 полей: FR (3 разряда), TLA (13 разрядов), NLA (24 разряда), SLA (16 разрядов), Interface ID (64 разряда)

в) Структура агрегированных адресов IPv6 включает 6 полей: FR (3 разряда), TLA (21 разряд), NLA (24 разряда), SLA (16 разрядов), Interface ID (64 разряда)

г) В структуре глобального агрегируемого уникального адреса в пакете IPv6 имеется 8 разрядов, зарезервированных для расширения префикса верхнего уровня.

10) Укажите номер правильного соответствия.

Код ответа сервера FTP	Описание
1 1yz	1 сервер начал требуемую операцию
2 2yz	2 сервер успешно закончил требуемое действие
3 3yz	3 сервер принял команду, но требуемое действие требует большего количества информации
4 x0z	4 ответ относится к ошибкам синтаксиса и синтаксически правильным командам, которые не попадают ни в одну функциональную категорию
5 x4z	5 ответ пока не определен
6 x5z	6 ответ информирует о состоянии файловой системы сервера

а) 1-1, 2-2, 3-3, 4-4, 5-5, 6-6

б) 1-2, 2-1, 3-3, 4-5, 5-4, 6-6

в) 1-1, 2-3, 3-2, 4-6, 5-5, 6-4

г) 1-3, 2-1, 3-2, 4-5, 5-6, 6-4

д) 1-4, 2-5, 3-6, 4-1, 5-2, 6-3

е) 1-4, 2-6, 3-5, 4-2, 5-1, 6-3

11) Массив поисковых образов Web-страниц- это:

а) FL- прямой список

б) IL- инвертированный список

в) индекс поисковой системы

г) таблица модифицированных страниц

12) Сколько управляющих соединений открывается по протоколу FTP?

а) одно

б) управляющее соединение не устанавливается

в) два

г) столько, сколько раз передаются данные

13) Укажите неверное утверждение:

128-разрядное пространство адреса (IPv6) использует:

а) иерархические глобальные одноадресные рассылки на основе адресов провайдеров

б) иерархические глобальные одноадресные рассылки по географическому признаку

в) личные адреса сайтов для использования только в пределах организации

г) локальные и глобальные многоадресные рассылки

д) широковещательные рассылки

14) Носитель, по которому взаимодействуют узлы на уровне связи данных (IPv6), называется:

а) пакетом (packet)

б) узлом (node)

в) связью (link)

г) соседями (neighbor)

15) Что означает число 826 рядом с названием группы новостей в окне перечня групп?

- а) число статей, прочитанных вами после подписки
- б) число участников группы новостей
- в) число статей, еще не прочитанных пользователем
- г) общее число статей на сервере новостей

16) Укажите номер неправильного ответа.

Команда FTP REIN (повторная инициализация):

- а) отменяет последнюю команду USER.
- б) сбрасывает все текущие операции ввода-вывода и информацию о доступе.
- в) прерывает выполнение текущей передачи.
- г) повторно устанавливает все параметры, используемые по умолчанию, но сохраняет управляющее соединение открытым.
- д) возвращает клиента к состоянию, в котором он находился сразу после установления управляющего соединения.

17) Узлы, подключенные к одной связи (IPv6), называются:

- а) пакетом (packet)
- б) узлом (node)
- в) соседями (neighbor)
- г) маршрутизатором (router)
- д) связью (link)

18) Укажите неверное утверждение:

- а) Заголовок CGI состоит из полей: Content-Type:, Location:, Status:
- б) Заголовок CGI состоит как из стандартных полей: Content-Type:, Location:, Status: , так и произвольных.
- в) Поле Status: позволяет вернуть CGI-скрипту статус обработки.
- г) Заголовок CGI состоит как из стандартных полей: Content-Type:, Location:, Status: Server: , так и произвольных.
- д) Поле Location: содержит URL ресурса, на который CGI-скрипт перенаправляет запрос

19) Какая строка не содержит переменных CGI-окружения:

- а) REQUEST\_METHOD, QUERY\_STRING
- б) CONTENT\_TYPE, REMOTE\_HOST
- в) SCRIPT\_NAME, SCRIPT\_FILENAME
- г) SERVER\_PROTOCOL, USER\_AGENT
- д) SERVER\_NAME, REMOTE\_ADDR
- е) SERVER\_PORT, SERVER\_SOFTWARE
- ж) REMOTE\_USER, SERVER\_PROTOCOL

20) Что из перечисленного ниже является путем по каталогам FTP?

- а) ftp.site.com
- б) docviewer.zip
- в) /pub/graphics/util
- г) ftp\pub\graphics\util

21) В приведенном ниже примере функция ereg () позволяет:

```

<?php
$referer=getenv("HTTP_REFERER");
$ip = getenv ("REMOTE_ADDR");
if (!ereg("http://fff.ru",$referer)){
exit("<center> $ip<br> $referer<br>OK");
}
///
?>

```

- а) проверить статус входа пользователя на сайт
- б) получить информацию, с какого адреса пришел пользователь
- в) извлечь информацию из переменных окружения
- г) получить информацию, по какому адресу пользователь входит на сайт
- д) нет правильных ответов

22) Укажите номер неправильного ответа.

- а) Команда FTP PORT 168,158,20,102,4,126 позволяет клиенту сообщать серверу на каком порту установить соединение для канала данных.
- б) Аргумент команды PORT - комбинация 32-битного IP-адреса компьютера и 16-битного адреса TCP-порта.
- в) С командой PORT задается аргумент, определяющий порт протокола для соединения данных.
- г) Клиент разбивает информацию аргумента команды PORT на части по 8 бит и передает каждое значение как десятичное число в кодировке ASCII.
- д) При работе на компьютере MSTUCA.RU (IP-адрес: 194.226.40.1) и использовании TCP- порта 1150 (0x047E) для получения данных необходимо подать команду: PORT 194.226.40.1 0x047E

23) Адрес типа unicast (IPv6):

- а) может быть назначен только интерфейсам маршрутизатора
- б) определяет группу интерфейсов
- в) определяет уникальный идентификатор отдельного интерфейса конечного узла или маршрутизатора
- г) ориентирован на применение маршрутизации от источника (Source Routing)
- д) идентифицирует группу интерфейсов, относящихся как правило, к разным узлам.

24) Опции FTP для операций по передаче файла, устанавливаемые по умолчанию:

- а) Nonprint, файловая структура, потоковый режим передачи
- б) ASCII, Nonprint, файловая структура, потоковый режим передачи
- в) ASCII, Nonprint, файловая структура
- г) Двоичный, Nonprint, файловая структура, потоковый режим передачи

25) К Telnet-клиентам относятся:

- |                        |                                      |
|------------------------|--------------------------------------|
| a) NCSA Telnet, TN3270 | б) NCSA Telnet, Eudora Light         |
| в) NCSA Telnet, TN3720 | г) TN3270, NCSA Telnet, Eudora Light |
| д) Eudora Light        | е) TN3270, TN3720                    |

26) Укажите неверное утверждение:

- а) SFTP поддерживает проверку прав пользователя (контроль доступа), передачу файлов, списки содержимого каталогов, изменение каталогов, переименование и удаление файлов.
- б) Протоколы SFTP и FTP используют транспортный протокол TCP.
- в) FTP, SFTP использует только одно TCP-соединение.
- г) SFTP использует только одно TCP-соединение

27) С помощью какого поля электронного письма можно выследить спамера и электронного злоумышленника:

- |              |                 |
|--------------|-----------------|
| а) Received  | б) Return Path  |
| в) From      | г) Message - IP |
| д) Encrypted | е) In-Reply-To  |

28) Информационной технологии как системе присущи основные признаки больших систем:

- а) наличие структуры; наличие единой цели функционирования; надежность; способность к развитию
- б) наличие единой цели функционирования; устойчивость к внешним и внутренним возмущениям; комплексный состав системы; способность к развитию, надежность системы
- в) наличие структуры; наличие единой цели функционирования; устойчивость к внешним и внутренним возмущениям; комплексный состав системы; способность к развитию
- г) наличие единой цели функционирования; агрегативность; надежность; комплексный состав системы; способность к развитию
- д) наличие структуры; наличие единой цели функционирования; способность к развитию

29) Приведенный способ отделений PHP от общего кода HTML `<? echo ( " test\n " ); ?>` соответствует:

- |                     |                  |
|---------------------|------------------|
| а) XML –документу   | б) ASP- стиль    |
| в) SGML -инструкции | г) для FrontPage |

30) К Intranet IP-numbers не относятся адреса:

- |   |                                      |
|---|--------------------------------------|
| а) 10.1.1.1 172.16.40.2 172.31.0.220100 | б) 10.1.1.1 172.17.40.2 192.168.24.2 |
| в) 172.24.5.4 192.168.26.3              | г) 172.15.8.7 10.2.40.5 192.168.4.3  |
| д) 10.10.10.10 172.29.30.6 192.168.4.3  | е) 10.0.0.1 172.30.40.5 192.168.0.1  |