МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ И
МИНИСТЕРСТВО ПО АТОМНОЙ ЭНЕРГИИ РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

ФАКУЛЬТЕТ “КИБЕРНЕТИКИ”

КАФЕДРА “КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ”

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К
ДИПЛОМНОМУ ПРОЕКТУ НА ТЕМУ:

Разработка системы управления хостингом для компаний-провайдеров.

Студент-дипломник: ___Нархов К.Г._____/ _____/
Руководитель проекта: ___Васильев Н.П._____ / _____ /
Рецензент: ___Коган Л.М._____ / _____ /
Заведующий кафедрой №12:___Забродин Л.Д._____ / _____ /

МОСКВА 2005 г.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ И ТЕРМИНОВ

ПРОВАЙДЕР - компания, предоставляющая клиентам услуги размещения данных на своих серверах, а также доступ к сети интернет.

ХОСТИНГ - услуга размещения данных клиента на сервере провайдером

ПО - программное обеспечение

ОС - операционная система

БД - база данных

СУБД - система управления базами данных

WWW - World Wide Web, всемирная паутина, одна из наиболее популярных служб сети Интернет

HTTP - Hyper Text Transfer Protocol, протокол передачи гипертекста, основной протокол службы WWW

HTML - Hyper Text Markup Language, единый стандартный язык форматирования документов

ASP - Active Server Pages, технология активных серверных страниц

IIS - Microsoft Internet Information Server, ПО для организации сервера служб сети Internet

Web-мастер - человек, отвечающий за функционирование Internet-сайта

Скрипт - интерпретируемый фрагмент программы

Браузер - программа просмотра страниц Интернет-сайтов

Введение

В настоящее время все большее и большее внимание уделяется проектам, связанным с размещением и последующим управлением данными в глобальной сети Internet. Internet дает целевую аудиторию для компаний, деятельность которых связанна с организацией быстрого и эффективного доступа к информации, в частности, интернет-провайдеров и поставщиком типичных для интернет услуг.

Данный дипломный проект посвящен разработке системы, которая позволяет в короткие сроки организовать на компьютере со стандартным набором ПО (входящим в ОС) полнофункциональный WWW- и FTP-сервер. То есть, данная система должна являть собой программное обеспечение для сервера сети, которое реализовывает процесс управления ресурсами и пользователями, а также предоставляет для этого удобные средства. Специальным требованием к разрабатываемой системе ставилось создание установочного пакета системы, чтобы иметь возможность дальнейшей быстрой установки системы на сервер без знания специфики данной процедуры.

Заметим, что данный проект не затрагивает таких специфических вопросов, как создание уникального дизайна и почтовой поддержки, а рассматривает исключительно вопросы создания и поддержки системного ПО сервера и web-приложения для удаленного управления.

Системное ПО – это программа (далее: ядро системы), запускаемая при каждом старте сервера, и набор административных скриптов. Ядро системы находится постоянно в памяти сервера и отслеживает поступающие запросы на выделение ресурсов сервера от web-приложения, кроме этого ядро системы предоставляет удобный интерфейс для администрирования БД.

Web-приложение – это программа, которая запускается на сервере и определяет реакцию, в частности, ответ сервера на пользовательские запросы. Таким образом, Web-приложение – это динамический сайт. В текущий момент технологии написания таких программ весьма разнообразны, это и скомпилированные программы: CGI и ISAPI -приложения – пример тому, и интерпретируемые и частично-интерпретируемые программы, здесь как пример можно привести технологии активных страниц ASP, PERL, PHP, JSP. Более подробный обзор этих средств будет проведен ниже.

Необходимость создания такой системы обуславливается всевозрастающим количеством компаний-провайдеров интернет услуг, использующих в своем серверном парке так называемы NT-машины, т.е. серверы с ОС от корпорации Microsoft – Windows 2000/2003 Server. Следует заметить, что web-приложение, разработанное в данном дипломном проекте, является масштабируемым и может быть использовано для удаленного управления NT-сервером в совокупности с серверным ПО других разработчиков. Ядро системы также не предусматривает обязательного использования web-приложение и может служить отдельным инструментом для управления ресурсами WWW- и FTP-сервера, функционирующего под Windows 2000/2003 Server. Таким образом, проблема, решенная в рамках данного дипломного проекта, оказалась весьма важной и актуальной, ввиду возможности разностороннего использования и возможности автоматизации большинства рутинных задач администратора NT-сервера.

Ядро системы отвечает за автоматизированное управления входящими в него административными скриптами, базой данных и стандартным DNS сервером.

Административные скрипты необходимы для настройки и управления MS IIS сервера и квотирования ресурсов пользователей.

Web-приложение – это совокупность сценариев пользователя и администратора, позволяющая пользователям сети Internet удаленно управлять их информацией, располагающейся на сервера, а также предоставляющая администраторам возможности эффективного управления ресурсами сервера:

- Администрирование пользовательских ресурсов;
- Администрирование дискового пространства сервера;
- Создание новых WWW- и FTP-узлов на сервере.

При создании данной системы учитывался опыт организации подобных систем на UNIX-серверах и был проведен анализ других систем удаленного управления серверами в сети Internet. На основании этого анализа выделены наиболее используемые функции и возможности, позволяющие развернуть данную систему на любом NT-сервере.

Сформулируем основные требования к такой системе:

1. Возможность и простота установки системы «с нуля»;
2. Использование базы данных для хранения информации системы;
3. Полнофункциональное управление ресурсами сервера: дисковым

пространством, пользователями и службами;

4. Возможность удобного администрирования системы;
5. Удобный интерфейс. Здесь имеется в виду функциональное удобство, но не удобство с эстетической точки зрения, что рассматривают вопросы оформления сайта (его дизайна).

Для достижения такой задачи необходимо детально проанализировать действующие системы автоматизирующие удаленное и локальное управление сервером, и выработать подробный перечень функций, которые данная система должна реализовывать. После такого анализа необходимо разработать базу данных для хранения информации системы. И, наконец, необходимо проанализировать возможности реализации системы и выбрать конкретное решение.

Так же весьма немаловажным аспектом проектирования системы является защита её ядра и web-приложения (административной его части особенно) от несанкционированного доступа. В рамках данного проекта будут рассмотрены так же варианты защиты Web-приложения от сетевого взлома.

1. Обзор современных СУБД и Web-технологий взаимодействия с ними

1.1 Обзор современных СУБД

На текущий момент технологии систем управления базами данных постоянно развиваются и совершенствуются. Поэтому сейчас параллельно существует огромное множество самых разнообразных СУБД. В данном обзоре затрагиваются только некоторые классические представители простых и сложных СУБД, но не затрагиваются такие последние разработки, как так называемые постреляционные СУБД, например, Caché. Причина этого в том, что использование данных СУБД для решения проблем, поставленных в данном дипломном проекте изначально невозможно как по конструкторским, так и по экономическим соображениям.

1.1.1 Простые СУБД

MS Access

Программный продукт MS Access представляет собой настольную систему управления базами данных (СУБД). Понятие "настольная" СУБД указывает на то, что все операции с базой данных осуществляются на локальном компьютере пользователя. Именно здесь находится физическое место хранения информации, а также работают средства управления и организации запросов. Программа успешно работает и в сетях, но с ограниченным числом клиентов. Сама компания Microsoft настоятельно рекомендует инсталлировать MS Access в рамках небольших рабочих групп. СУБД прекрасно справляется с такими типовыми задачами, как учет складского хозяйства, обеспечение работы магазина и т. д. На Access вполне можно написать продвинутую бухгалтерскую или кадровую программу. С таким же успехом он применим и в качестве служебного средства, например, для поддержки сложного Web-сайта или системы intranet. Вообще, MS Access позволяет создать сколь угодно сложный по своей архитектуре продукт. Вопрос лишь в масштабах его применения: чем больше пользователей одновременно должно работать с базами данных и чем плотнее осуществляются транзакции, тем ближе граница между настольным MS Access и более мощными СУБД, например MS SQL Server.

MySQL

СУБД MySQL также как и Access является небольшим, компактным и простым в использовании сервером баз данных. MySQL доступна на ряде платформ Unix, Windows NT/2000, 95/98. Основные плюсы таковы: быстродействие; безопасность; открытость кода, доступность (разработчики предоставляют программное обеспечение бесплатно через Интернет и осуществляют бесплатную поддержку и консультации); надежность; поддержка поставщиком Интернет-услуг (провайдером); простые процедуры загрузки и выгрузки данных.

Недостатками данных СУБД является отсутствие транзакций, триггеров, хранимых процедур, невысокая производительность. Данные СУБД являются превосходным решением для небольших интернет-сайтов, в первую очередь, конечно MySQL. В данном случае важным недостатком является необходимость установки дополнительного ПО на сервер.

1.1.2 Высокопроизводительные СУБД

SQL Server

С самого начала SQL Server 2000 разрабатывалась в качестве составной части Windows 2000 DNA, анонсированной как многофункциональная интегрированная платформа для разработки и эксплуатации распределенных Web-приложений и Web-сервисов нового поколения. В основе новой версии СУБД лежит хорошо зарекомендовавшее себя ядро SQL Server 7.0, существенно переработанное для удовлетворения растущих аппетитов разработчиков ПО для Интернет. Одним из наиболее интересных нововведений в SQL Server 2000 является поддержка языка XML на уровне самой СУБД. Другим важным усовершенствованием в новой СУБД является возможность объединения с помощью Web существующих многомерных кубов OLAP, что значительно упрощает анализ бизнес-информации. Активное использование SQL Server 2000 в электронной коммерции и Web предполагает рост требований к системе безопасности. Понимая это, разработчики Microsoft приложили немало усилий для повышения надежности защиты информации. В руках администратора появилась более гибкая и мощная система безопасности на основе ролей, определяющих права доступа к серверу, базе данных и приложениям. Кроме того, существенно усилены средства аудита — появилась возможность отслеживания 18 типов классов, связанных с безопасностью, и дополнительных подклассов. Теперь администратор может отслеживать не только успешные или

неудачные попытки установления соединения с SQL Server, но и более детально — доступ к объектам, их создание и удаление, изменение прав доступа и т. д. Шифрование в SQL Server 2000 поддерживается и на уровне сетевого трафика, и на уровне файлов баз данных, что является серьезным улучшением.

Существенным минусом системы является то, что SQL Server поддерживает только платформу Windows NT. Одним из преимуществ SQL Server является простота его применения, в частности администрирования. SQL Server Enterprise Manager, входящий в состав всех редакций Microsoft SQL Server (за исключением MSDE), представляет собой полнофункциональное и достаточно простое средство для администрирования этой СУБД.

Oracle

Последние три вышедшие версии Oracle носят общее официальное название — Oracle8i (в маркетинге они позиционируются как СУБД для Internet). Oracle8i Release 1 (версия 8.1.5) была выпущена в 1999 году, Release 2 — в начале 2000-го, Release 3 — в конце прошлого года. Их предшественница, Oracle8, была выпущена в 1998 году, но в мире до сих пор имеется немало пользователей Oracle7: многим из них просто не нужна поддержка больших баз данных и объектных типов, отличающая Oracle8 от Oracle7, равно как и новшества, появившиеся в Oracle8i. Для поддержки Internet в Oracle8i существенно расширено применение языка Java. В частности, в состав серверной части этой СУБД входит виртуальная Java-машина, в дополнение к имевшимся ранее средствам выполнения кода на языках SQL и PL/SQL. Oracle Enterprise Manager в последних версиях Oracle8i содержит разделяемый репозитарий, который позволяет координировать доступ пользователей к данным, хранящимся в различных серверах Oracle, с помощью Oracle Management Server — компонента среднего звена, предназначенного для централизации управления доступом клиентов к серверам. В Oracle8i впервые была применена концепция Virtual Private Database для упрощения управления доступом: теперь средства защиты данных встроены в саму базу данных, а не в приложения. Большим минусом данной СУБД является ее высокая рыночная стоимость, а также сложность администрирования.

DB2

Пакет DB2 выпускается в двух редакциях: DB2 Workgroup и DB2 Enterprise Edition. В данной СУБД реализованы все известные по предшествующим версиям DB2 новаторские технологии механизма БД, такие, как распараллеливание обработки запроса, полный набор средств тиражирования, сводные таблицы запросов для повышения производительности БД, возможности объектно-ориентированного конструирования баз данных и средства языка Java. К этому надо добавить, что система DB2 оснащена полым набором мультимедиа-расширений, позволяющих сохранять текст, звук и видео- фрагменты, изображения и географические данные и манипулировать ими. Можно говорить, что по возможностям масштабирования разработанная специалистами IBM технология кластеризации баз данных не имеет аналогов. Эти расширения существенно облегчают процесс разработки приложений для Web, а так же программ, содержащих фотоизображения и объемные текстовые отчеты. Система DB2 вполне конкурентоспособна и в качестве платформы для разработки приложений, т.к существует средство Stored Procedure Builder - автоматически преобразовывающее оператор SQL в соответствующий класс Java и включающее его в структуру базы данных. В версии DB2 6.1 значительно улучшена функциональная совместимость с другими СУБД: пакет позволяет использовать разработанную Microsoft спецификацию OLE DB, новый стандарт доступа к базам данных. Средства административного управления СУБД DB2, которые в новой версии переписаны на Java и могут быть получены из Web, заслуживают самой высокой оценки. Одним из основных минусов данной СУБД является сложность ее администрирования. Кроме того, для IBM DB2 поддержка Windows NT и процессоров Intel не является в такой же степени приоритетной задачей, как для Oracle и Microsoft, поэтому и использование их СУБД на данной платформе вряд ли оправдано с точки зрения перспективы.

1.2 Web-технологии Microsoft: ISAPI и ASP

В основном современные технологии программирования под интернет можно классифицировать как:

- Универсальные
- Специализированные

В первому типу можно отнести технологию CGI, которая, впрочем, может быть по-разному реализована для каждой операционной системы. Наиболее известные средства разработка, основывающиеся на CGI – языки PHP и Perl.

К специализированным средствам можно отнести технологии, разработанные корпорацией Microsoft под собственную операционную систему Windows, - ISAPI и ASP. Хотя в настоящий момент технология ASP имеет реализацию под ОС UNIX.

ISAPI

Ведущие производители программного обеспечения для Web-серверов решили обеспечить непосредственную связь с Web-сервером для технологий программирования, которые придут на смену CGI. Новая стандартизированная концепция была названа ISAPI (Internet Server Application Programming Interface - прикладной программный интерфейс Internet-сервера). Эта же технология для серверов Netscape называется NSAPI (Netscape Server Application Programming Interface). Используя эти технологии, можно писать программы обработки запросов и динамического формирования страниц на языке С и других языках программирования общего назначения. В отличие от CGI-программ, программы ISAPI не выгружаются из оперативной памяти сервера после выполнения запроса. Выполняя те же операции, что и CGI-програмы, они избавлены от необходимости выполнения дополнительных операций, связанных с загрузкой и выгрузкой в начале и конце обработки каждого запроса.

Microsoft предлагает ISAPI-программы двух различных типов: фильтры, которые перехватывают запросы до их поступления на сервер, и приложения, обрабатывающие Web-запросы. Официально, ASP считается приложением ISAPI. Однако приложение ASP следует отличать от приложений, написанных для ASP. В книге под приложением ASP мы будем подразумевать движок ASP (ASP engine). Приложение ISAPI ASP представляет собой библиотеку динамической компоновки (DLL) (asp. dll), которая выполняет синтаксический

ISAPI-приложения задействуются *после того,* как Web-сервер обнаружит запрос. Они позволяют повысить возможности сервера при сохранении остальных его функций. Движок ASP представляет собой ISAPI-приложение. Сервер US (Internet Information Server - информационный сервер Internet) передает запрос ASP-файлов движку ASP, который его обрабатывает и динамически формирует возвращаемую страницу.

ISAPI-фильтры срабатывают *прежде,* чем Web-сервер обнаруживает запрос. Они позволяют осуществлять мониторинг или перехват запросов для специальной обработки. Например, программа классификации запросов и распределения полосы пропускания сервера на основе IP-адресов пользователей - неплохой кандидат в ISAPI-фильтры. Она будет перехватывать и фильтровать запросы до их передачи на Web-сервер. Кроме того, фильтр можно использовать в качестве брандмауэра (firewall) для блокирования запросов от источников, не заслуживающих доверия.

ASP

В основе технологии Active Server Pages (ASP) лежит очень простая идея - совмещение в узлах Web HTML-текста с программным кодом. Мы сможем убедиться, что такое совмещение дает превосходные результаты. Имея все достоинства мощного Web-сервера, ASP обрабатывает запросы пользователей и возвращает индивидуализированные страницы, создаваемые динамически на основе логики и информации, хранящейся в файлах и базах данных, но, кроме этого, обрабатывает индивидуализированные пользовательские данные. Другими словами, ASP предоставляет возможность одновременного выполнения программы на сервере некоторым количеством пользователей. При этом очень существенна возможность доступа к большой ЭВМ (мэйнфрейму), наряду с возможностями обработки графики локальными рабочими станциями. Все это объединено средством просмотра, способным работать в разных платформах (браузером), и встроено в коммуникационный механизм мирового стандарта (Web). В этой книге мы покажем, как воспользоваться этими богатыми возможностями для создания гибких, индивидуализированных приложений. ASP имеет ряд преимуществ по сравнению с большинством языков разработки Web-приложений, особенно при разработке приложений для intranet:

Размещение ASP-программы в текстовых файлах. Текстовые файлы легко модифицировать даже после установки программы. Огромное

преимущество - возможность устранения неисправности с удаленного компьютера при помощи текстового редактора. Поддержка и модификация Web-приложений на основе компилированного кода и регистрируемых объектов ActiveX намного сложнее. К тому же, что очень важно, текстовую программу, написанную сегодня, вполне можно модифицировать в текстовом редакторе завтра: отсутствует зависимость от интегрированной среды разработки (IDE). Со средствами разработки, генерирующими код, дело обстоит иначе: они постоянно совершенствуются и часто меняются. Обычно (хотя и не всегда) обеспечивается совместимость с более ранними версиями, но применение иных сред разработки, как правило, невозможно. В ASP для внесения изменений в программу достаточно текстового редактора.

Ограничение времени выполнения ASP-программы. По умолчанию, выполнение ASP-программы прекращается через 90 секунд после запуска. Как правило, предпочтителен более короткий интервал. Ограничение времени позволяет легко справляться с ситуациями зацикливания либо запроса миллиона записей из базы данных - задержка в работе сервера не превысит установленного интервала. Многие Internet-провайдеры (ISP), использующие компилированные приложения, начнут со временем применять ASP именно благодаря ограничению времени выполнения.

Безопасность ASP-программ. ASP-программа выполняется в ограниченном пространстве. Например, в ASP нельзя обычным образом выполнить чтение или запись двоичного файла. Полное зависание US-сервера на оригинальных ASP-сценариях крайне маловероятно, если вообще возможно. Это еще одна причина для ISP выбирать ASP-приложения после отказа от других технологий.

Небольшой объем ASP-приложений. Поскольку все DLL уже установлены на сервере, для запуска ASP остается лишь установить файлы с программами, изображениями и средствами поддержки. Эти файлы, как правило, имеют небольшой объем и могут быть эффективно сжаты. Следует помнить, что по мере добавления к приложению компилированных компонентов, выигрыш в объеме снижается.

1.3 Общие CGI-технологии: Perl и PHP

CGI-программы - это небольшие выполняемые файлы, запускаемые сервером в ответ на запросы от браузера. CGI-программа обрабатывает запрос и возвращает браузеру данные в формате HTML. Однако, в отличие от статической HTML-страницы, данные, которые возвращает такая программа, могут формироваться в зависимости от отправителя, места отравления, времени суток и множества других условий. К счастью, компания Microsoft согласилась с тем, что CGI-программы - не лучшее средство динамического формирования страниц. Причина недостаточной их эффективности -необходимость загрузки для обработки каждого отдельного запроса.

Объяснение кроется в недрах протокола HTTP (HyperText Transfer Protocol - протокол передачи гипертекстовых файлов). HTTP-запрос - это краткая транзакция между клиентом и сервером. После завершения транзакции клиент и сервер забывают друг о друге. Представим себе двух людей, приглашенных на свадьбу. При встрече они знакомятся, затем расходятся и тут же забывают друг друга. Подобная ситуация была в кинофильме *"День сурка".* Каждый день сменяется не новым, а все тем же днем, и герои фильма вынуждены изо дня в день повторять свои действия. Главный герой, которого играет Билл Муррей - единственный, кто осознает ситуацию. Именно таково положение с Web, хотя там и нет Билла Муррея.

Поскольку и для клиента, и для сервера каждая HTTP-транзакция - событие единственное и неповторимое, она начинается со сложного процесса квитирования. Более того, CG[-программа, выполнявшая предыдущий запрос, оказывается выгруженной, и для обработки следующего запроса ее приходится загружать вновь. При этом на загрузку и выгрузку программы времени, зачастую, тратится больше, чем на выполнение. Для Web-сервера, который должен работать с максимальной эффективностью, такая ситуация неприемлема.

PERL

Язык Perl был создан в 1986 году как инструмент для администрирования и конфигурирования системных ресурсов в сети, состоящей из Unix-компьютеров. Постепенно Perl (чья аббревиатура расшифровывается как *Практический Язык для Извлечения текстов и Генерации отчетов* (Practical Extraction and Reporting

Language)) эволюционировал в межплатформенный язык и оказался в центре внимания процветающего кибернетического сообщества.

Perl является интерпретируемым языком, предназначенным для сканирования текстовых файлов, извлечения из них информации и вывода, на основе полученных таким образом данных, текстовых отчетов. То есть программа **perl** (обратите внимание на отсутствие заглавной буквы) используется для выполнения сценариев Perl. (Хотя компиляторы Perl тоже существуют.) В этой главе мы начнем изучение сценариев Perl.

Некоторые люди удивляются популярности Perl (языка, ориентированного на текстовый ввод и вывод и запускаемого из командной строки) в мире графических интерфейсов типа Windows. Популярность Perl продолжает расти по ряду причин.

Многие операционные системы остаются текстово-ориентированными.

- Perl является межплатформенным языком, максимально идентично поддерживаемым в разных операционных системах, и отличается только в нескольких неизбежных деталях (таких, как число байтов, используемых для представления длинного целого).
- На самом деле Perl *обладает* определенными графическими возможностями за счет взаимодействия с популярным модулем **Tk.pm**. Данный модуль позволяет использовать стандартные графические интерфейсные элементы (widgets) с помощью вспомогательного средства — библиотеки Tk языка Tel. Это позволяет создавать из Perl окна с кнопками, меню и другими объектами.
- Однако с точки зрения явного большинства программистов текущая популярность Perl подпитывается программированием Common Gateway Interface (CGI-программированием), применяемого для операций по взаимодействию клиент/сервер в среде Web. Когда речь идет о создании Web-страниц, текстовая ориентированность языка перестает быть недостатком, так Как они также являются чисто текстовыми объектами. CGI-программирование на Perl представляет собой очень мощный инструмент, и, соответственно, это одна из основных тем, которые мы будем рассматривать.

PHP

PHP скрипты используются на массе серверов в Интернет. В ноябре 2000 г. на сайте http://nevod.ru/linux был проведен опрос на тему "На каком языке проще,

быстрее и качественнее писать CGI?". В опросе приняло участие 954 респондента (по состоянию на 28.11.00). По результатам опроса PHP оказался на втором месте (после Perl) по популярности среди языков, используемых для создания CGI-скриптов. Perl набрал 334 голоса (35 процентов), а PHP - 232 голоса (24 процента). Все остальные языки, включая C, C++, Delphi и Visual Basic, набрали не более 11 процентов голосов. Конечно, тут надо сделать поправку на то, что опрос проводился на сайте, посвященном ОС Линукс, а значит, результаты опроса могут не совсем правильно отражать реальную картину использования разных языков, но все же эти результаты говорят о том, что PHP достаточно широко используется.

PHP - это язык программирования для динамической генерации Web-страниц с помощью скриптов, запускаемых на Web-сервере. Вы создаете страницу с помощью PHP and HTML. Когда посетитель сайта открывает страницу, сервер выполняет включенные в html-код операторы PHP и посылает результат браузеру посетителя, точно так же, как это делается с помощью ASP или ColdFusion. Однако, в отличие от ASP или ColdFusion, PHP является продуктом с открытым исходным кодом (Open Source) и платформенно-независим. PHP работает на Windows NT и многих версиях Unix. Он может быть запущен как модуль в Apache или как исполняемая программа через CGI. В случае запуска в виде модуля Apache, PHP работает особенно легко и быстро. В этом случае отсутствуют накладные расходы, связанные с созданием процессов, поэтому результат выдается быстро, и не требуется настраивать mod_perl для уменьшения расходования памяти сервера.

PHP предоставляет великолепные возможности по доступу к базам данных (и ODBC), а также интеграцию с различными внешними библиотеками. Это позволяет Вам делать все - от генерации PDF-документов до грамматического разбора в XML.

Операторы PHP вставляются прямо в Ваши Web-страницы, так что нет необходимости в специальных средах разработки (IDE). Синтаксис языка PHP подобен синтаксису C и Perl. Вы не должны объявлять переменные до их использования. PHP даже имеет какие-то зачатки объектно-ориентированного программирования, обеспечивая удобный способ организации кода и его инкапсуляции. Хотя PHP быстрее всего работает в том случае, когда он встроен в Apache, на Web-сайте PHP имеются инструкции по его установке на Microsoft IIS и Netscape Enterprise Server.

2. Анализ средств управления web-сервером MS IIS 5.0

В MS IIS 5.0 доступные следующие возможности, необходимые для корректного функционирования сервиса хостинга.

Перезапуск IIS. Теперь IIS может быть перезапущен без необходимости перезагрузки компьютера.

Создание резервной копии и восстановление данных IIS. Можно создать резервную копию и сохранить установки метабазы для упрощения возврата в безопасное (известное) состояние.

Возможности настройки. Разрешения «Чтение», «Запись», «Выполнение», «Сценарий» и «FrontPage Web» могут быть установлены на уровне узла, каталога или файла.

Наблюдения за активностью узла: диаграммы, представляющие в режиме реального времени статистические сведения по интенсивности обмена данными, такие как число запросов за день, число запросов за час, число посетителей за день и число посетителей за час.

Программные возможности: полная поддержка ASP, включая высокопроизводительные компоненты ASP и новые возможности обработки ошибок.

Централизованное администрирование: Административные средства IIS используют Microsoft® Management Console (MMC). Консоль содержит программы (называемые оснастками), которые используются администраторами для управления своими серверами. Администрировать компьютер, на котором в системе Windows 2000 Server запущена служба IIS, в интрасети с компьютера, управляемого операционной системой Windows 2000 Professional, позволяет оснастка IIS.

Защита приложения: IIS 5.0 предлагает более мощную защиту и увеличивает надежность веб-приложений. По умолчанию IIS будет выполнять все приложения в общем или *групповом* процессе, который отделен от процессов ядра IIS. Кроме того, остается возможность *изолирования* критически важных приложений, которые следует запускать и вне процессов ядра IIS, и вне группового процесса.

2.1 Общее представление об web- и ftp-узлах в MS IIS 5.0

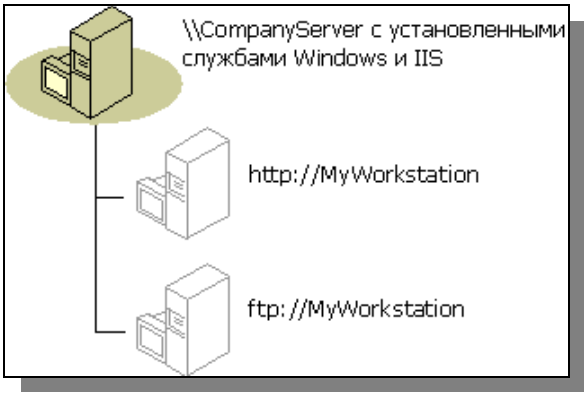


Рис 1. WWW и FTP узлы IIS сервера.

В настоящее время несколько веб- или FTP-узлов могут одновременно выполняться на одном компьютере с операционной системой Windows 2000, производя впечатление нескольких работающих компьютеров. Каждый веб-узел способен поддерживать несколько имен доменов. Поскольку каждый узел выглядит как отдельный компьютер, узлы иногда называют *виртуальными серверами*. В Windows 2000 Professional можно использовать компьютер для размещения одного веб-узла и одного FTP-узла. В приведенном ниже примере администратор интрасети установил IIS на рабочую станцию подразделения и создал два узла, один из которых является веб-узлом , а второй узлом FTP. Если необходимо разместить несколько веб- или FTP-узлов на одном компьютере, необходимо перейти на версию Windows 2000 Server.

Свойства — это значения, которые могут быть установлены на веб-узле. Свойства узла отображаются в окне свойств и хранятся в базе данных, называемой *метабазой*.

Во время установки IIS различным свойствам присваиваются значения по умолчанию, которые отображаются в окне свойств. Можно использовать значения, предлагаемые IIS по умолчанию, или настроить эти параметры для удовлетворения потребностей публикации в Интернете. Корректировка стандартных настроек иногда позволяет добавить новые функциональные возможности, повысить производительность и усилить систему безопасности.

Свойства могут задаваться на уровне узла, на уровне каталога или на уровне файла. Параметры, заданные на верхнем уровне, таком как уровень узла, автоматически используются, т.е. *наследуются* на нижних уровнях, например, на уровне каталога. Кроме того, значения этих параметров могут изменяться по отдельности на нижнем уровне. Если изменить свойство для отдельного узла, каталога или файла, то изменение настройки основных свойств не приведет автоматически к переопределению значения свойства, заданного явно на нижнем

уровне. Вместо этого будет выведено сообщение с приглашением подтвердить, следует ли изменить свойство отдельного узла, каталога или файла соответственно новому значению основного свойства.

Значения некоторых свойств имеют вид списка. Например, значение документа, используемое по умолчанию, может быть списком документов, которые будут загружаться, если пользователь в адресе URL не указал имя файла. Другими примерами свойств, сохраняемых в формате списков, являются специальные сообщения об ошибках, управление доступом TCP/IP, сопоставления сценариев и сопоставления MIME. Хотя эти списки обычно содержат несколько элементов, IIS рассматривает весь список как единое значение свойства. Если пользователь изменяет список для каталога, а затем вносит глобальное изменение в свойства на уровне узла, то список на уровне каталога полностью заменяется на новый список с уровня узла; объединение списков не производится. Кроме того, значения свойств, имеющих формат списка, отображаются только на уровне основных свойств, а также на уровне узла или каталога, на котором эти свойства были изменены относительно значения по умолчанию. Списки не отображаются, если они представляют унаследованные значения по умолчанию.

Фильтры ISAPI (см. главу 1) отображаются в формате списков, но не обрабатываются как списки. Если добавить фильтры на уровне узла, то новые фильтры добавляются к списку фильтров с уровня основных свойств. Если два фильтра имеют одинаковый приоритет, то фильтр с уровня основных свойств загружается раньше, чем фильтр с уровня узла.

Перезапуск FTP связан к потерей сетевого соединения при загрузке файлов. Клиенты, поддерживающие перезапуск FTP должны только переустановить FTP-соединение с помощью команды REST. Передача файлов возобновится автоматически с того места, где она была прервана.

Кроме того, применение перезапуска FTP в IIS 5.0 невозможно при использовании FTP для загрузки запросов с подстановочными знаками (MGET), выгрузке файлов на сервер (PUT), или загрузке файлов, больших 4 гигабайт.

2.2 Записи системного реестра для MS IIS 5.0

Записи реестра для служб WWW:

AcceptByteRanges (тип REG_DWORD, диапозон 0,1. По умолчанию включен)

Этот параметр определяет, будет ли сервер HTTP обрабатывать заголовок "Range" для типа "bytes:". Если этот параметр включен, сервер подает сигнал о том, что он принимает запросы в диапазоне, отправляя поле заголовка "Accept-Range: bytes", после чего обрабатывает поступающий запрос с полем заголовка "Range: bytes=" согласно проекту Интернета «Byte range extension to HTTP».

AllowSpecialCharsInShell (тип REG_DWORD, диапозон 0,1. По умолчанию отключен) Это значение определяет, являются ли специальные символы программы Cmd.exe [| (, ; % < >] разрешенными для командной строки при запуске пакетных файлов (.bat и .cmd). Использование таких символов связано с серьезным риском для системы безопасности. Если для данного параметра задано значение 1, то злонамеренные пользователи получают возможность выполнения команд на сервере. Таким образом, настоятельно рекомендуется оставлять для этого параметра заданное по умолчанию значение 0. Стандартное значение параметра не позволяет передавать специальные символы в минипрограммы CGI, сопоставленные сценариям. Если задать значение 1, то становится возможной передача специальных символов в минипрограммы CGI, сопоставленные сценариям, за исключением символа вертикальной черты | и стандартных символов перенаправления ввода-вывода < и >, имеющих специальный смысл для командного процессора.

DLCCookieNameString (тип REG_STRING. Строка. По умолчанию не используется) Этот параметр задает имя модуля настройки узла HTTP (Cookie), которое сервер отправляет клиентам нижнего уровня. Отправленный модуль настройки выполняет роль заменителя заголовка HOST, что позволяет серверу направлять запросы HTTP клиента на соответствующий веб-узел.

DLCHostNameString (тип REG_STRING. Строка. По умолчанию не используется) Этот параметр определяет имя веб-узла, который содержит меню хост-компьютера нижнего уровня, которое сохраняется в параметре **LCCookieMenuDocumentString**. Меню хост-компьютера нижнего уровня является документом (например, файлом HTML, .asp и т.д.), в котором перечислены веб-узлы, имеющие общий IP-адрес. Из этого меню посетитель выбирает

соответствующий экземпляр сервера.

DLCMungeMenuDocumentString (тип REG_STRING. Строка. По умолчанию не используется) Этот параметр указывает имя файла меню хост-компьютера для клиентов, которые не поддерживают модули настройки узла (cookies). Этот файл используется для детализации адресов URL, запрошенных клиентом нижнего уровня путем вставки имени хост-компьютера в адрес URL.

SSIEnableCmdDirective (тип REG_DWORD, диапозон 0,1. По умолчанию отключен) Директива **#exec cmd** на стороне сервера включает выполнение команд оболочки. На узлах с защищенным содержимым рекомендуется в качестве дополнительной меры безопасности отключить директиву **#exec cmd**, задавая для данного параметра значение 0, в особенности, когда посторонним разрешается размещать файлы на сервере. Этот параметр не существует в реестре по умолчанию. Чтобы разрешить этой директиве выполнение команд оболочки, следует сначала создать параметр и задать для него значение 1.

UploadReadAhead (тип REG_DWORD, диапозон 0-0x80000000. По умолчанию 48 Кб) При отправке клиентом данных на сервер, этот параметр задает количество данных, которое сервер читает по умолчанию перед передачей управления в приложение. После этого ответственным за чтение оставшихся данных становится приложение. Если требуется увеличить это значение, следует увеличить объем оперативной памяти сервера.

UsePoolThreadForCGI (тип REG_DWORD, диапозон 0,1. По умолчанию включен) IIS по умолчанию использует группы потоков на сервере для выполнения обработки сценариев CGI. Это означает, что запросы CGI в течение длительного промежутка времени могут исчерпать группу потоков сервера.

Записи реестра для служб FTP:

AnnotateDirectories (тип REG_DWORD, диапозон 0,1. По умолчанию отключен)

Служба FTP поддерживает аннотации каталогов с помощью специальных сообщений. Текст аннотации сохраняется в специальном файле с именем ~ftpsvc~.ckm в каталоге, для которого создается аннотация. Если такой файл существует в каталоге назначения операции смены каталога (CWD) FTP, то служба возвращает в операцию содержимое этого файла. Это позволяет администраторам добавлять специальные сообщения для обрабатываемых каталогов. По умолчанию в конфигурации службы отправка текста аннотации не

задается. Если предполагается добавлять специальные сообщения, необходимо создать файл аннотации. Кроме того, рекомендуется сделать файл аннотации скрытым, чтобы он не отображался в списке файлов каталога.

EnablePortAttack (тип REG_DWORD, диапозон 0,1. По умолчанию отключен)

Значение этого параметра задается по умолчанию, чтобы предотвратить возможность нарушения безопасности в спецификации протокола FTP. В спецификации службы FTP допускается установление пассивных подключений на основании адреса порта, задаваемого клиентом. Это позволяет компьютерным хулиганам выполнять в службе FTP разрушительные команды. Эта проблема возникает, когда подключение к службе FTP выполняется с помощью порта, отличного от порта данных FTP (20), номер которого меньше, чем IP_PORT_RESERVED (1024). Параметр **EnablePortAttack** определяет, возможность такой атаки на компьютер. Служба по умолчанию не выполняет подключения к портам, номера которых меньше IP_PORT_RESERVED (кроме 20). Если требуется обеспечить возможность пользователям подключаться с помощью других портов, как указано в документе FTP RFC, этот параметр следует включить.

LowercaseFiles (тип REG_DWORD, диапозон 0,1. По умолчанию отключен)

Служба FTP использует исходный регистр символов в именах файлов (как эти имена сохраняются в файловой системе). Однако для совместимости с файловыми системами, в которых имена анализируются с учетом регистра, может оказаться необходимым использование имен в фиксированном регистре. После добавления значения этого параметра администраторами, служба будет при сравнении имен использовать имена с символами нижнего регистра.

2.3 Метабаза IIS 5.0

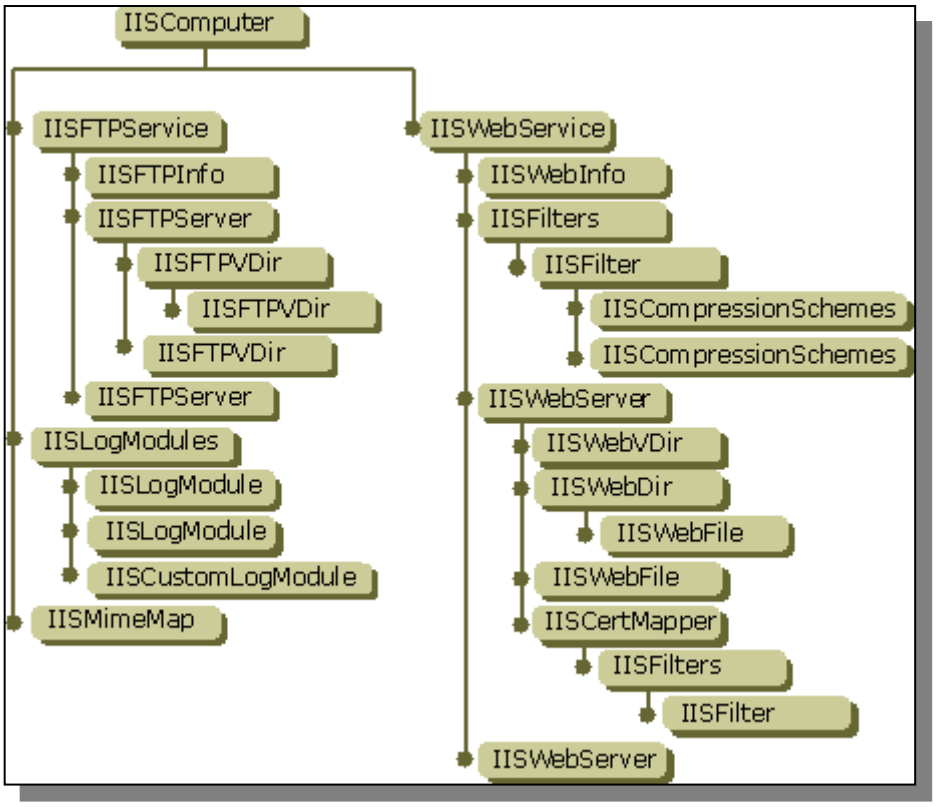


Рис 2. Общая структура метабазы IIS.

Метабаза модифицируется путем настройки значений с помощью таких административных средств, как оснастка IIS и диспетчер служб Интернета (HTML). Кроме того, метабазу можно изменять и программным способом, используя объекты IIS Admin Objects и IIS Admin Base Object.

Неверная настройка свойств метабазы может привести к неполадкам, в том числе к сбоям в работе веб- или FTP-узла. Ошибочная настройка может повлечь повреждение конфигурации узла. В метабазе следует изменять только те свойства, которые нельзя настроить с помощью оснастки IIS или диспетчера служб Интернета (HTML); при непосредственном редактировании метабазы следует соблюдать осторожность.

Пространство имен

Пространство имен определяет местоположение свойств метабазы. Оно организовано следующим образом:

LM/*Служба*/*веб-узел*/Root/*виртуальный каталог*/*каталог*/*файл*

где LM = Local Machine

Служба = служба Интернета (W3SVC или MSFTPSVC)

веб-узел = имя веб-узла

Root = виртуальный корневой каталог

виртуальный каталог = имя виртуального каталога

каталог = имя каталога

файл = имя файла

IIsWebVirtualDir

Объект IIsWebVirtualDir служит для установки свойств метабазы, относящихся к одному или всем виртуальным каталогам веб-узла. При использовании объекта IIsWebVirtualDir в корневом каталоге сервера (.../W3SVC/2/ROOT) значения наследуемых свойств будут применены ко всем виртуальным подкаталогам. Можно устанавливать свойства для определенного виртуального каталога, используя объект IIsWebVirtualDir для определенного виртуального каталога (.../W3SVC/2/ROOT/*виртуальный_каталог*).

Для создания и веб-приложений в виртуальных веб-каталогах и подкаталогах и управления ими также служат методы IIsWebVirtualDir. Приложения также можно определять и управлять ими в веб-каталогах.

Объект IIsWebVirtualDir является объектом-контейнером ADSI.

ADsPath

Для корневого виртуального каталога сервера:

IIS://*имя_компьютера*/W3SVC/*N*/ROOT

где *имя_компьютера* может быть любым именем или строкой "LocalHost".

Для определенного виртуального каталога:

IIS://*имя_компьютера*/W3SVC/*N*/ROOT/*виртуальный_каталог*

где *имя_компьютера* может быть любым именем или строкой "LocalHost".

2.4 Средства управления DNS серверами в ОС Windows 2000/2003 Server

Служба DNS, выполняющаяся в операционной системе Windows 2000, обеспечивает следующие возможности.

- **Сервер имен DNS в соответствии с документами RFC**

  DNS представляет открытый протокол и стандартизируется с помощью документов RFC. Microsoft поддерживает эти спецификации и удовлетворяет им.

- **Взаимозаменяемость с другими реализациями DNS-сервера**

  Поскольку служба DNS является совместимой со спецификациями RFC и может использовать стандартные файлы данных DNS и форматы записей ресурсов, она может успешно работать со многими реализациями DNS-серверов, например, с реализациями, использующими программное обеспечение BIND (Berkeley Internet Name Domain).

- **Поддержка службы каталогов Active Directory**

  DNS требуется для поддержки службы каталогов Active Directory. Если установить службу каталогов Active Directory на сервере, операционная система Windows 2000 Server может автоматически установить и задать конфигурацию DNS-сервера, если таковой не обнаруживается в текущей конфигурации сервера.

  Сначала в мастере установки Active Directory следует указать имя DNS домена Active Directory, для которого выполняется повышение сервера до контроллера домена. Далее в процессе установки мастер выполняет следующие проверки.

  1. На основании конфигурации клиента TCP/IP проверяется, имеется ли в конфигурации основной DNS-сервер, подходящий для использования.
  2. Если основной DNS-сервер доступен, выполняется запрос для поиска главного полномочного сервера домена DNS, указанного ранее при работе мастера.
  3. Затем проверяется, может ли главный полномочный сервер поддерживать и принимать динамические обновления.
  4. Если в этот момент невозможно обнаружить поддерживающий DNS-сервер, который должен принять обновления для указанного

доменного имени DNS, используемого в службе каталогов Active Directory, пользователь получает возможность установить сервер, который использует службу DNS-сервер.

В общем, использование службы DNS-сервер Windows 2000 настоятельно рекомендуется для обеспечения интеграции и поддержки службы каталогов Active Directory, а также расширенных возможностей DNS-серверов. Однако можно использовать другой тип DNS-сервера для поддержки развертывания службы каталогов Active Directory.

При использовании других DNS-серверов следует рассмотреть дополнительные вопросы, относящиеся к взаимозаменяемости DNS.

- **Интеграция с другими сетевыми службами Microsoft**

Служба DNS допускает интеграцию с другими службами Windows 2000 и содержит средства, выходящие за рамки спецификаций RFC. Эти средства включают интеграцию со службами Active Directory, WINS и DHCP.

- **Удобство администрирования**

Оснастка DNS обеспечивает усовершенствованный графический интерфейс пользователя для управления службой DNS. Кроме того, операционная система Windows 2000 Server содержит несколько мастеров конфигурации, позволяющих выполнить общие задачи администрирования сервера. Кроме оснастки DNS имеются другие инструменты, помогающие в управлении и поддержке DNS-серверов и клиентов в сети.

- **Поддержка протокола динамического обновления согласно спецификациям RFC**

Служба DNS позволяет клиентам динамически обновлять записи ресурсов на основании протокола динамического обновления DNS (RFC 2136). Это улучшает характеристики администрирования DNS за счет уменьшения времени, которое ранее требовалось на обновление этих записей вручную. Компьютеры, выполняющие Windows 2000, могут динамически регистрировать свои имена DNS и IP-адреса.

- **Поддержка добавочных зонных передач между серверами**

Зонные передачи между DNS-серверами используются для репликации информации о части пространства имен DNS. Добавочные зонные передачи используются для репликации только изменившихся частей зоны, что позволяет рациональнее использовать пропускную способность сети.

2.6 Средства управления DNS клиентами в ОС Windows 2000/2003 Server

Служба DNS-клиент, выполняющаяся в Windows 2000, используется для сопоставления доменных имен DNS и реализует следующие возможности.

- **Системное кэширование**

  Записи ресурсов из ответов на запросы добавляются в кэш клиента по мере запросов приложений к DNS-серверам. Эта информация сохраняется в кэше в течение заданного срока жизни и может использоваться для ответов на последующие запросы.

- **Поддержка кэширования отрицательных ответов согласно документам RFC**

  В дополнение к кэшированию утвердительных ответов от DNS-серверов (содержащих информацию записей ресурсов в ответе на запрос), служба DNS-клиент кэширует также отрицательные ответы. Отрицательный ответ является результатом запроса, когда не существует запись ресурса для запрашиваемого имени.

  Кэширование отрицательных ответов позволяет предотвратить повторение запросов к несуществующим именам, что может неблагоприятно повлиять на быстродействие клиента. Любая отрицательная информация запросов кэшируется на более короткое время, чем используется для утвердительных ответов; по умолчанию в течение не более 5 минут. Это исключает продолжение кэширования устаревших отрицательных ответов, если записи через некоторое время становятся доступными.

  Кэширование отрицательных ответов является новой спецификацией стандарта DNS, которая определена в документе RFC 2308. Дополнительные сведения см. в этом документе.

- **Прекращение обращений к не отвечающим DNS-серверам**

Служба DNS-клиент использует список серверов, упорядоченный по предпочтениям. Этот список включает все основные и дополнительные DNS-серверы, указанные в конфигурации каждого из активных сетевых подключений компьютера.

1.3 Организация пространства доменных имен DNS

Любое доменное имя DNS в дереве технически представляет домен. Однако в большинстве дискуссий DNS имена идентифицируются одним из пяти способов на основании уровня и способа использования имени. Например, доменное имя DNS, зарегистрированное для корпорации Майкрософт (microsoft.com.), представляет домен второго уровня. Это имя состоит из двух частей (называемых метками), показывающих, что оно находится на втором уровне сверху от корня или вершины дерева. Большинство доменных имен DNS содержат две или большее число меток, каждая из которых задает новый уровень в дереве. Точки используются в именах для разделения меток.

Кроме доменов второго уровня в следующей таблице представлены другие термины, используемые для описания доменных имен DNS по их функциям в пространстве имен.

Табл. 1. описания доменных имен DNS

Тип имени	**Описание**	**Пример**
Корень доменов	Вершина дерева, представляющая неименованный уровень; иногда обозначается парой прямых кавычек (**""**), указывающих пустое значение. При использовании в доменном имени DNS устанавливается с помощью завершающей точки (**.**), обозначающей, что имя расположено в корне или на самом верхнем уровне иерархии доменов. В данном случае доменное имя DNS рассматривается как полное и указывает на точное расположение в дереве имен. Имена, установленные таким способом, называют полными доменными именами (FQDN).	Единственная точка (**.**) или точка, использованная в конце имени, например, «example.microsoft.com.».
Домен верхнего	Имя из двух или трех букв, которое используется, чтобы указать	«.com» указывает имя, зарегистрированное

уровня	страну/регион или тип организации.	для коммерческого использования в Интернете.
Домен второго уровня	Имена переменной длины, зарегистрированные для индивидуальных пользователей или организаций для использования в Интернете. Эти имена всегда базируются на соответствующем домене верхнего уровня в зависимости от типа организации или географического расположения, в котором используется имя.	«microsoft.com.» является именем домена второго уровня, зарегистрированным для корпорации Майкрософт регистратором доменных имен DNS Интернета.
Поддомен	Дополнительные имена, которые организация может создавать как производные от зарегистрированного имени домена второго уровня. Такие имена обеспечивают рост дерева имен DNS в организации и его распределение по отделам или по географическому расположению.	«example.microsoft.com.» представляет имя несуществующего поддомена, предназначенного для примеров имен в документации корпорации Майкрософт.
Имя узла или ресурса	Имя, представляющее лист в дереве имен DNS, которое определяет конкретный ресурс. Обычно крайняя левая метка в доменном имени DNS определяет конкретный компьютер в сети. Например, имя этого уровня, используемое в записи ресурса узла (A), используется для поиска IP-адреса компьютера по его имени узла.	«host-a.example.microsoft.com.», где первая метка («host-a») представляет имя узла DNS для конкретного компьютера в сети.

2.8 Управление общими свойствами, полномочиями и ресурсами DNS

При добавлении зоны с помощью консоли DNS доступны общие свойства, определяющие следующие возможности:

- пауза или запуск зоны для прерывания или восстановления службы;
- изменение или преобразование типа зоны;
- отключение или включение динамического обновления зоны.

Для зон, интегрированных в службу каталогов, можно включить использование только безопасных обновлений. Это позволяет ограничить обновления только набором авторизованных пользователей или компьютеров. Когда для зоны включены безопасные обновления, только пользователям, компьютерам или группам, авторизованным через службу каталогов Active Directory и включенным в список управления доступом (ACL) для каждой интегрированной зоны, разрешается обновлять зону или используемые в ней специфические записи ресурсов.

В дополнение к общим свойствам зоны с консоли DNS можно задавать или изменять следующие свойства.

- **Свойства начальной записи зоны** Включают свойств, поддерживаемые начальной записью зоны, которая используется для инициализации зоны и указывает полномочия зоны для доменного имени DNS (вместе с любыми поддоменами, не делегированными на другие серверы) для других членов пространства имен DNS. Эта запись определяет, как часто зона должна обновляться и передаваться другими серверами, которые загружают зону, а также сколь долго клиенты могут кэшировать записи ресурсов при возвращении ответов на запросы имен в зоне.

- **Свойства записи сервера имен** Включают все поля, поддерживаемые записью ресурса NS для зоны. Запись ресурса NS используется для назначения полномочных DNS-серверов зоны.

- **Свойства зонных передач** Средство задать конфигурацию, определяющую, как выполняются зонные передачи.

Имеется возможность отклонять все запросы к серверу на передачу этой зоны, разрешить передачи только на DNS-серверы, заданные на

вкладке **Серверы имен**, а также передавать зону только на DNS-серверы, указанные по IP-адресам в списке.

Кнопка **Уведомить** позволяет включить и настроить отправку уведомлений DNS на дополнительные серверы зоны. Когда используются уведомления, другие серверы (либо указанные на вкладке **Серверы имен**, либо включенные в список) получают уведомления об изменениях зоны. Эти серверы могут затем запрашивать изменения, инициируя зонную передачу для обновления зоны.

- **Свойства просмотра WINS** Средство просмотра WINS (Windows Internet Name Service) позволяет использовать расширенный путь сопоставления имен DNS для зон, когда запрошенное имя не обнаружено в зоне. Если для зоны включен просмотр WINS, то можно обратиться к WINS-серверу (или списку WINS-серверов) за помощью в сопоставлении имени узла в управляемом WINS пространстве имен NetBIOS. Эта возможность поддерживается только DNS-серверами Microsoft.

Управление зонами основывается на концепции полномочий сервера. Когда DNS-сервер настроен на загрузку зоны, он использует записи ресурсов двух типов для определения свойств, определяющих полномочия для зоны.

- Вначале запись ресурса начальной записи зоны (SOA) указывает имя происхождения зоны и содержит имя сервера, который является основным источником информации для зоны. Кроме того, эта запись указывает другие основные свойства зоны.

- Далее запись ресурса сервера имен (NS) используется для обозначения DNS-серверов, назначенных полномочными серверами для зоны. После включения сервера в запись ресурса NS он становится известным для других как полномочный сервер для этой зоны. Это означает, что любой сервер, указанный в записи ресурса NS, должен рассматриваться остальными как полномочный источник и может отвечать на запросы, относящиеся к именам, включенным в эту зону.

Записи ресурсов SOA и NS играют особую роль в конфигурации зоны. Эти записи являются обязательными для любой зоны и обычно являются первыми записями ресурсов, перечисленными в файлах. Мастер добавления новой зоны по умолчанию автоматически создает эти записи, когда новая основная зона добавляется с помощью оснастки DNS.

Запись ресурса начальной записи зоны (SOA)

Запись ресурса начальной записи зоны (SOA) всегда является первой в любой стандартной зоне. Она указывает DNS-сервер, который либо изначально был сделан, либо был в процессе работы определен как основной сервер зоны. Эта запись также используется для сохранения других свойств, таких как сведения о версии и временных параметрах, затрагивающих обновление и истечение срока существования зоны. Эти свойства определяют способ выполнения зонных передач между серверами, которые являются полномочными серверами зоны.

Запись ресурса NS

Записи ресурсов серверов имен (NS) используются для назначения указанным серверам полномочий для доменного имени DNS двумя способами:

- с помощью списка полномочных серверов для домена, который делает эти серверы известными для других компьютеров, запрашивающих информацию о домене (зоне);
- путем указания полномочных DNS-серверов для всех поддоменов, которые делегируются из зоны.

Когда назначаются серверы с именами узлов в той же зоне, соответствующие записи ресурсов адресов (A) обычно используются в зоне для сопоставления имен указанных серверов их IP-адресам. Для серверов, которые указаны с помощью этой записи ресурса при делегировании зоны в поддомен, запись ресурса NS обычно содержит имена вне зоны. Чтобы сопоставить имена вне зоны, могут оказаться необходимыми записи ресурсов A для указанных серверов вне зоны. Такие записи NS и A вне зоны, обеспечивающие делегирование, называют *связывающими записями*.

После создания зоны в нее следует добавить дополнительные записи ресурсов. Обычно добавляются следующие записи ресурсов.

- **Узел (A)** Для сопоставления доменного имени DNS с IP-адресом, используемым компьютером.
- **Псевдоним (CNAME)** Для сопоставления псевдонима доменного имени DNS с другим первичным или каноническим именем.

- **Почтовый обменник (MX)** Для сопоставления доменного имени DNS с именем компьютера, который выполняет обмен или перенаправление почты.
- **Указатель (PTR)** Для сопоставления обратного доменного имени DNS, основанного на IP-адресе компьютера, указывающего на прямое доменное имя DNS этого компьютера.
- **Расположение службы (SRV)** Для сопоставления доменного имени DNS с указанным списком узлов DNS, предлагающих определенный тип службы, например, с контроллерами домена службы каталогов Active Directory.
- Другие записи ресурсов по мере необходимости.

Записи ресурсов узлов (A)

Записи ресурсов (A) не являются обязательными для всех компьютеров, но они необходимы для компьютеров, совместно использующих ресурсы в сети. Любой компьютер, совместно использующий ресурсы в сети, который должен идентифицироваться по своему доменному имени DNS, нуждается в записях ресурсов A, которые обеспечивают сопоставление имени DNS с IP-адресом компьютера.

Большая часть записей ресурсов A, требуемых в зоне, может относиться к другим рабочим станциям и серверам, содержащим общие ресурсы, другим DNS-серверам, почтовым серверам и веб-серверам. Такие записи ресурсов представляют большинство записей ресурсов в базе данных зоны.

Записи ресурсов псевдонимов (CNAME)

Записи ресурсов псевдонимов (CNAME) иногда называют *каноническими именами*. Такие записи позволяют использовать несколько имен для указания на один узел, что облегчает одновременное использование одного компьютера в качестве FTP-сервера и веб-сервера. Например, общеизвестные имена сервера (ftp, www) регистрируются с помощью записей ресурсов CNAME, которые обеспечивают сопоставление с именем узла DNS, таким как "server-1", для компьютера, на котором выполняются эти службы.

При переименовании компьютера с существующей записью ресурса A в зоне имеется возможность использовать временную запись ресурса CNAME, чтобы предоставить пользователям и программам отсрочку для переключения от использования старого имени компьютера к использованию нового имени.

Когда запись ресурса CNAME используется для создания псевдонима или переименования компьютера, следует задать лимит времени на использование этой записи в зоне перед ее удалением из DNS. Если пользователь забывает удалить запись ресурса CNAME, а затем удаляется соответствующая запись ресурса A, наличие записи CNAME может привести к напрасному расходованию ресурсов сервера на попытки сопоставления в запросах имени, которое больше не используется в сети.

Обычно и чаще всего запись ресурса CNAME требуется для создания постоянного псевдонима доменного имени DNS при сопоставлении универсальных имен, базирующихся на имени службы, таких как www.example.microsoft.com, нескольким компьютерам или IP-адресам, используемым на веб-сервере. Например, ниже демонстрируется общий синтаксис использования записи ресурса CNAME.

псевдоним **IN CNAME** *первичное_каноническое_имя*

Записи ресурсов почтового обменника (MX)

Запись ресурса почтового обменника (MX) используется приложениями электронной почты для обнаружения почтового сервера по доменному имени DNS, используемому в адресе получателя сообщения электронной почты. Например, запрос DNS к имени «example.microsoft.com» может использоваться для поиска записи ресурса MX, что позволит приложению электронной почты направлять или обмениваться сообщениями с пользователем, имеющим почтовый адрес «user@example.microsoft.com».

Запись ресурса MX показывает доменное имя DNS для компьютера или компьютеров, которые обрабатывают почту для домена. Если существуют несколько записей ресурсов MX, служба DNS-клиент пытается установить связь с почтовыми серверами в порядке указанного предпочтения, от минимального значения (высший приоритет) к максимальному (низший приоритет). Ниже приводится пример основного синтаксис при использовании записи ресурса MX.

Записи ресурсов указателя (PTR)

Записи ресурсов указателя (PTR) используются для поддержки процесса обратного просмотра, который выполняется на основании зон, создаваемых в корневом домене in-addr.arpa. Такие записи используются для обнаружения компьютера по его IP-адресу и сопоставления этой информации с доменным именем DNS этого компьютера.

Записи ресурсов PTR могут добавляться в зону следующими способами.

- Имеется возможность вручную создать запись ресурса PTR для статического клиентского компьютера TCP/IP с помощью оснастки DNS, либо в отдельной процедуре, либо в процессе создания записи ресурса A.
- Компьютеры, выполняющие Windows 2000, могут использовать службу DHCP-клиент для динамической регистрации и обновления собственных записей ресурсов PTR в DNS, когда конфигурация IP изменяется.
- Для всех других клиентских компьютеров с включенной службой DHCP записи ресурсов PTR регистрируются и обновляются DHCP-сервером, если эти компьютеры получают аренду IP от квалифицированного DHCP-сервера. Такую возможность предоставляет служба DHCP, обеспечиваемая операционной системой Windows 2000 Server.

Записи ресурса указателя (PTR) используется только в зонах обратного просмотра для поддержки процесса обратного просмотра.

Записи ресурсов размещения службы (SRV)

Чтобы обнаружить контроллеры домена службы каталогов Active Directory в Windows 2000, требуются записи ресурсов расположения службы (SRV). Обычно нет необходимости администрировать вручную записи ресурсов SRV при установке службы каталогов Active Directory.

Мастер установки службы каталогов Active Directory по умолчанию пытается обнаружить DNS-сервер по списку основных или дополнительных DNS-серверов, указанных в свойствах клиента TCP/IP для каждого из его активных сетевых подключений. Если выполняется соединение с DNS-сервером, который может принимать динамическое обновление записи ресурса SRV (и других записей ресурсов, относящихся к регистрации Active Directory как службы в DNS), то процесс задания конфигурации завершается.

Если во время установки не обнаруживается DNS-сервер, который может принимать обновления для доменного имени DNS, используемого службой каталогов Active Directory, то DNS-сервер Windows 2000 может быть установлен локально и автоматически настроен с зоной, базирующейся на домене службы каталогов Active Directory.

Например, если доменом Active Directory, выбранным в качестве первого домена в лесу, является example.microsoft.com, будет добавлена зона с корневым доменным именем DNS example.microsoft.com. Эта зона будет настроена на использование с DNS-сервером, выполняющемся на новом контроллере домена.

Если не установить DNS-сервер, обеспечиваемый Windows 2000, то в процессе установки службы каталогов Active Directory создается и записывается файл (Netlogon.dns), содержащий записи ресурсов SRV и другие записи ресурсов, требуемые для поддержки Active Directory. Этот файл создается в папке %SystemRoot%\System32\Config.

Если используется DNS-сервер, отвечающий одному из следующих описаний, необходим использовать записи в файле Netlogon.dns, чтобы вручную настроить основную зону на этом сервере для поддержки службы каталогов Active Directory.

- Компьютер, работающий как DNS-сервер, выполняет другую операционную систему, такую как UNIX, и не может распознавать или принимать динамические обновления.
- DNS-сервер на этом компьютере является полномочным для основной зоны, соответствующей доменному имени DNS домена службы каталогов Active Directory.
- DNS-сервер поддерживает записи ресурсов SRV, определенные в предварительном Интернет-документе «A DNS RR specifying the location of services (DNS SRV)», но не поддерживает динамические обновления.

Например, служба DNS, обеспечиваемая Windows NT Server 4.0, после обновления Service Pack 4 или более поздней версии удовлетворяет этому описанию.

В будущем запись ресурса SRV может также использоваться для регистрации и поиска общеизвестных служб TCP/IP в сети, если приложения реализуют и поддерживают запросы имен DNS, указывающие записи этого типа.

3. Разработка программного комплекса для управления сервисом хостинга

Разработка сервиса хостинга состояла из четырех основных этапов. Сначала был разработан скрипт WHS (kernel.js), который реализовывал все необходимые функции, определенные для сервиса в техническом задании. Особенностью скрипта, являлось то обстоятельство, что он должен был запускаться локально под правами администратора системы. Приемлемое, но не оптимальное решение. Вторым этапом явилась разработка клиентского ISAPI приложения, которое смогло бы при обращении к нему извне запустить под правами администратора созданный на первом этапе скрипт. Напомним, что система безопасности IIS устроена таким образом, что программы запущенные веб-сервером при обращении пользователя обладают весьма ограниченными правами, недостаточными для полнофункциональной работы kernel.js. После перебора всех доступных средств, было принято решение разбить ядро сервиса на клиент-серверную подсистему. Клиентом автоматически стало ISAPI приложение, а сервер был создан на заключительном этапе разработки. Четвертым этапом явилась разработка приложения (в составе серверной части), которое бы отвечало за настройку локальной DNS службы.

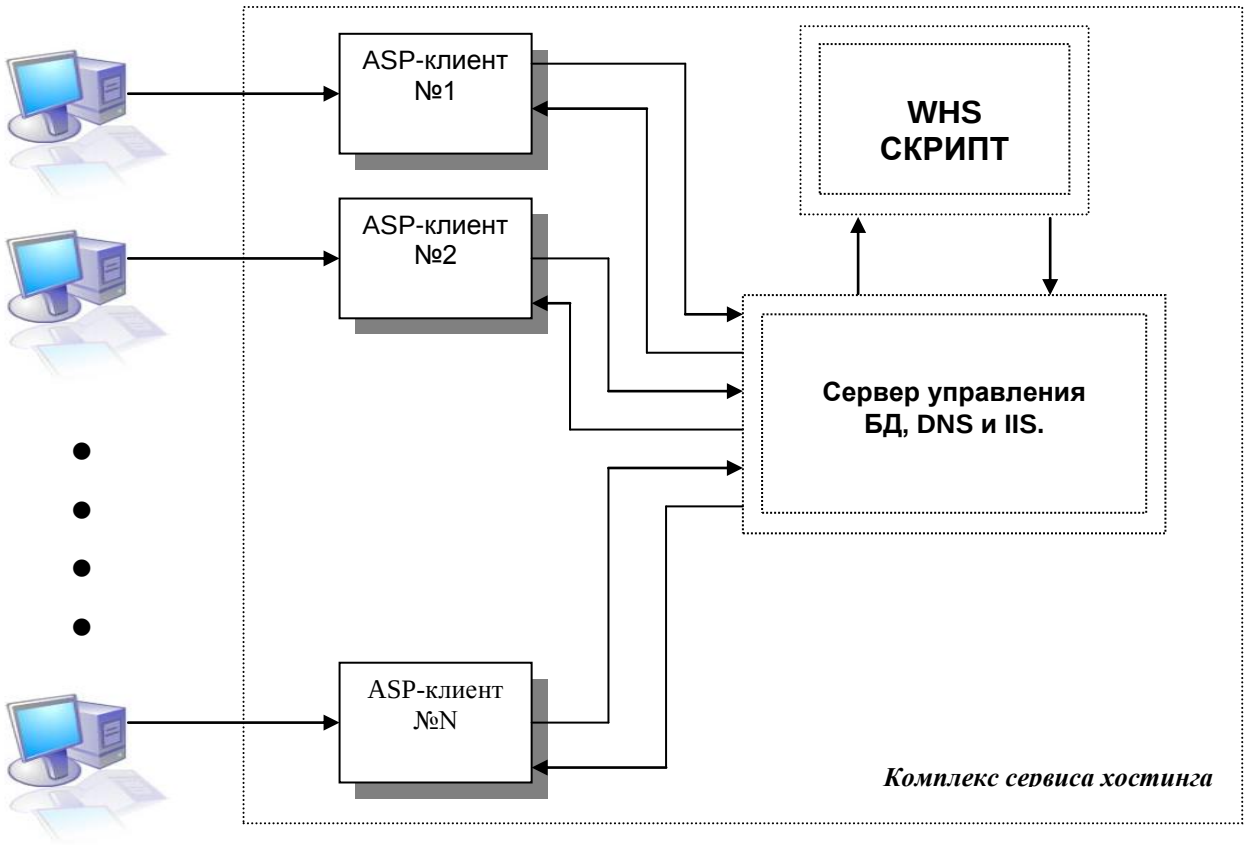


Рис 3. Обобщенная схема системы

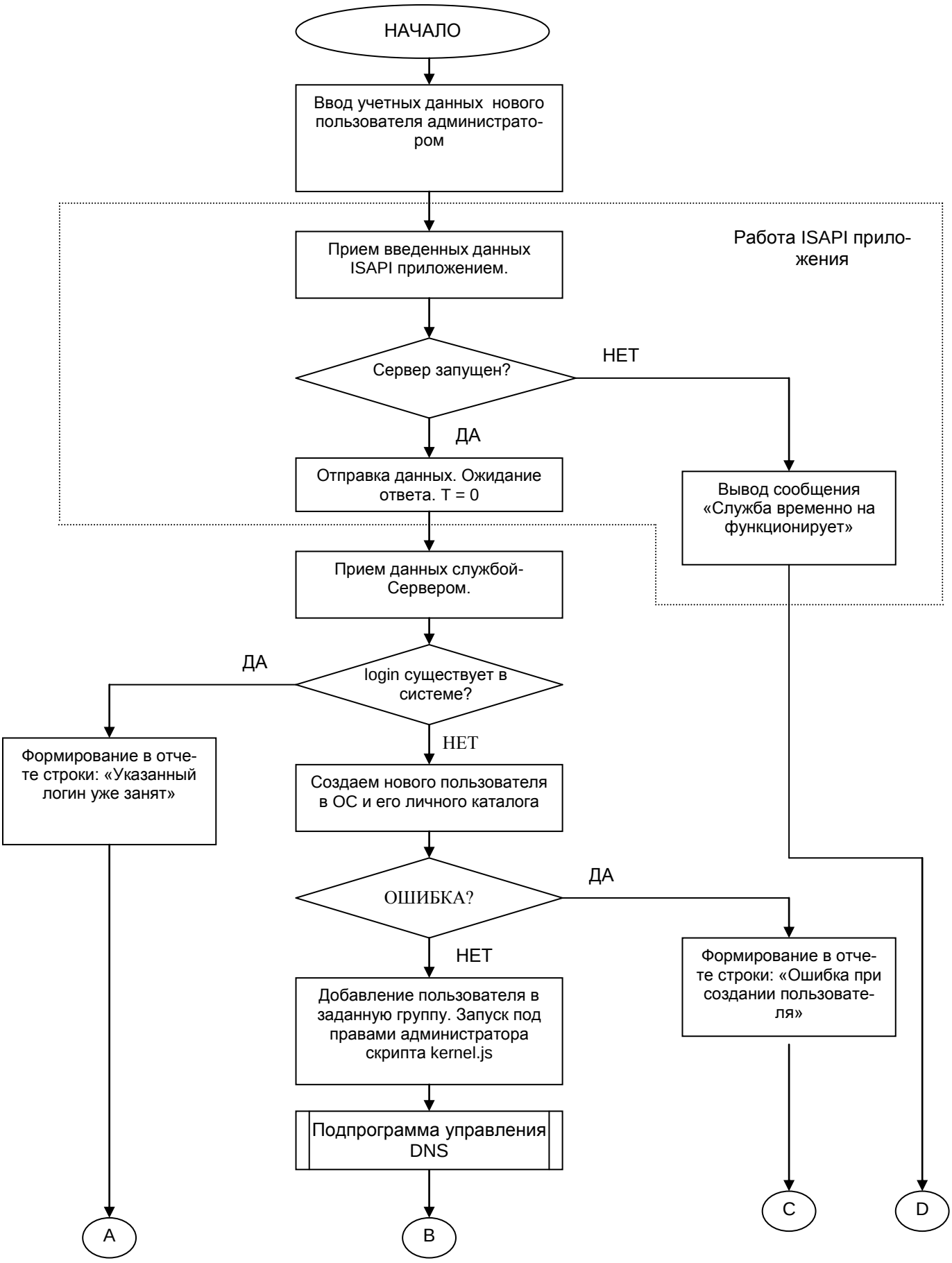
3.1 Разработка ПО подсистемы управления пользователями и ресурсами

В техническом задании были поставлены следующие требования к разрабатываемой системе. Они обуславливались, в первую очередь, операционной системой, под управлением которой должен работать сервис, а также стандартным набором функций, присутствующих на аналогичных системах в интернете.

Следует заметить, что в рамках данного дипломного проекта был реализован полный набор необходимых функциональных возможностей:

- Система должна управляться через Ядро системы. Управляющая программа должна иметь форму, которую заполняет администратор. На основе отправленных программой данных сервис регистрирует сайт.
- На основе данных, полученных от управляющей программы, должна быть создана учетная запись в ОС сервера и базе данных системы. То есть в систему должен быть добавлен пользователь с логин-псевдонимом и паролем, которые были указаны при заполнении формы.
- Должна быть обеспечена проверка на присутствие пользователя с заданным логин-псевдонимом в системе.
- Необходимо задать квоту на дисковое пространство, созданного пользователя. Сервис предоставляет каждому участнику не более 5242880 байт дискового пространства на сервере.
- На основе введенного логин-псевдонима должны быть созданы одноименный www-узел на запущенном сервере IIS и домен, в котором создаются узлы.
- Взаимодействие с DNS должно быть обеспечено на уровне файлов (%SYSTEM DIR%\dns*.dns). Кроме того, Ядро системы, взаимодействующее с DNS, должно быть оснащено функцией перезапуска (запуск, останов) службы доменных имен.
- Должна быть обеспечена проверка на присутствие узлов с таким же именем на запущенном IIS сервере.
- Доступ к созданным узлам должен быть обеспечен в течении кротчайшего промежутка времени после регистрации.

3.1.1 Общий алгоритм работы комплекса



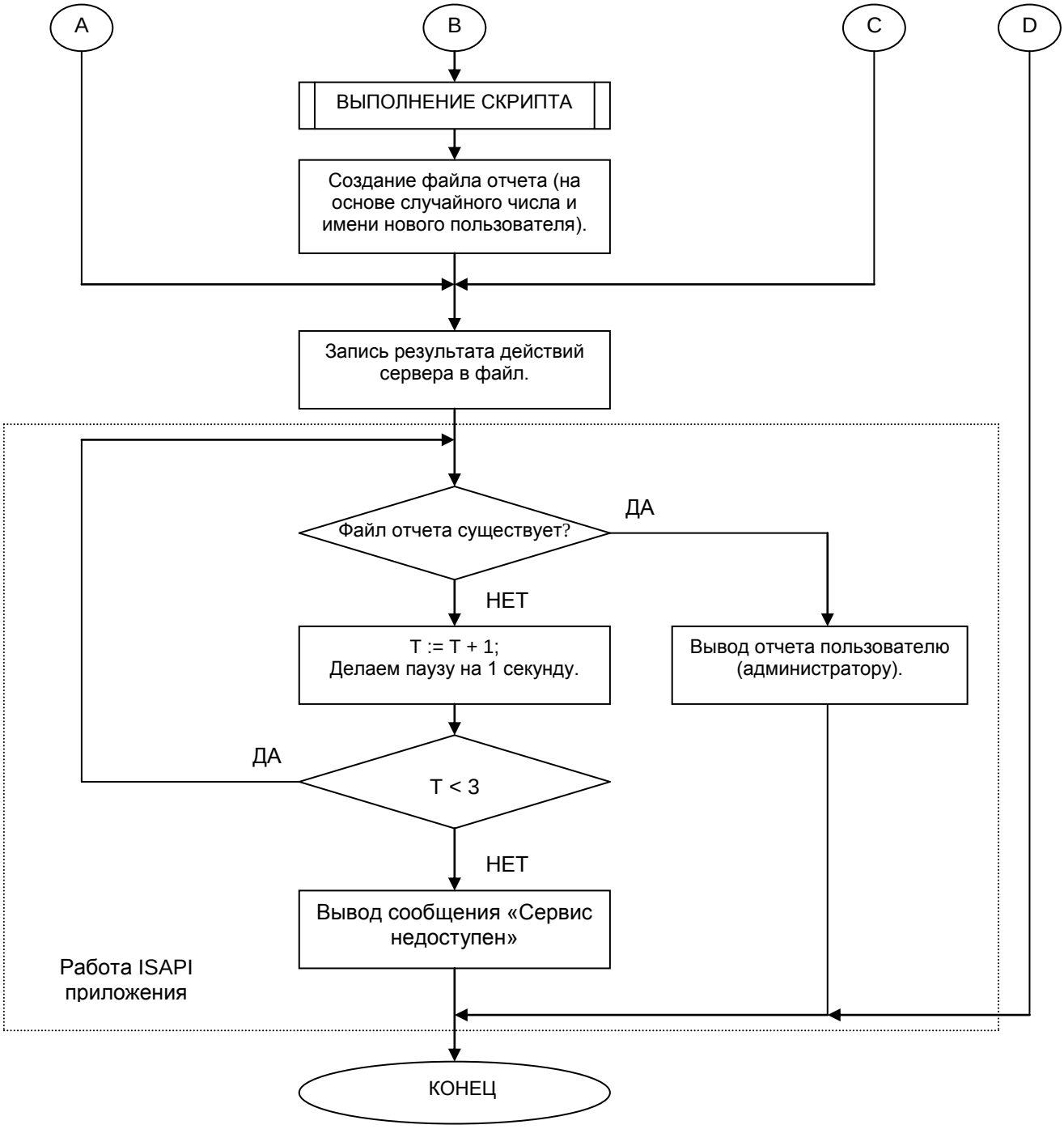


Рис 4. Общий алгоритм работы комплекса

3.1.2 Решение проблем безопасности

Безопасность системы клиент – сервер.

Данный сервис имеет ряд преимуществ и недостатков с точки зрения безопасности. В дипломном проекте недостатки сервиса, выявленные в предыдущих учебно-исследовательских работах, устранены. Перечислим сперва достоинства:

- Наиболее уязвимым местом любого интренет комплекса являются так называемые «шлюзовые» скрипты или приложения, которые принимают введенные пользователем данные. В случае нашего сервиса таковым является клиентское ISAPI-приложение. Связано это с тем, что ISAPI клиент – откомпилированная программа со строго определенным набором функций. Также данная программа имеет статический алгоритм, не зависящий от формата пользовательского ввода. Данный ISAPI клиент не работает с файловой системой сервера «на запись» и выводит на экран (в стандартный ISAPI-вывод) только данные отчета, находящиеся в специальном файле. Эти особенности исключают возможность изменения файловой системы сервера и вывод на экран пользователю конфиденциальные данные.
- Технология ISAPI обеспечивает быструю работу клиентского приложения и полную совместимость со средой работы IIS сервера. Имеется ввиду то обстоятельство, что ISAPI – технология, созданная непосредственно под IIS, и как следствие наиболее эффективная при использовании на данном сервере.

Недостатки:

- Слабое место в данной системе то, что клиент и сервер связываются между собой посредством механизма сокетов. Оба сокета, тем не менее, работают только адресом 127.0.0.1, тем не менее всегда “слушающий” сокет сервера является потенциальной целью злоумышленника.
- При зависании какого-либо из компонентов системы (клиента или сервера) сервис становится недоступным в целом.
- Система работает под известной своей “ненадежностью” ОС Windows.

Решения проблемы безопасности IIS.

Частично эта проблема описана в введении к данной главе. Дело в том, что применение сокетов для связи клиента с сервером придумано не случайно. В Windows 2000 существует ряд привилегий и политик безопасности, распределенных на уровне пользователей и групп. В принципе, дело обычное для любой серверной ОС. MS IIS (веб-сервер, под управление которого работает наш сервис) все дочерние процессы (IIS – ветвящийся сервер) запускает с привилегией, которая не дает процессу никаких прав на управление учетными даннымя ОС. Это вполне ожидаемо и закономерно. Тем не менее, нам по техническому заданию необходимо создать в ОС новую учетную запись и виртуальную директорию в IIS. Для этого нужны права как минимум администратора. В процессе разработки было опробовано много путей получения администраторских прав в рамках ISAPI процесса, однако подходящего решения найдено не было, поэтому и был реализован дополнительный сервер, работающий под административными правами. Возник следующий вопрос: а как связать между собой клиент и сервер? Первым решением была OLE/DCOM технология, но, к сожалению, она не сработала. Вызов COM-сервера производился ISAPI приложением в собственном контексте, в связи с чем все права и привилегии наследовались. Единственным удачным решением оказалась проверенная технология сокетов.

3.1.3 Алгоритмы работы клиента и сервера

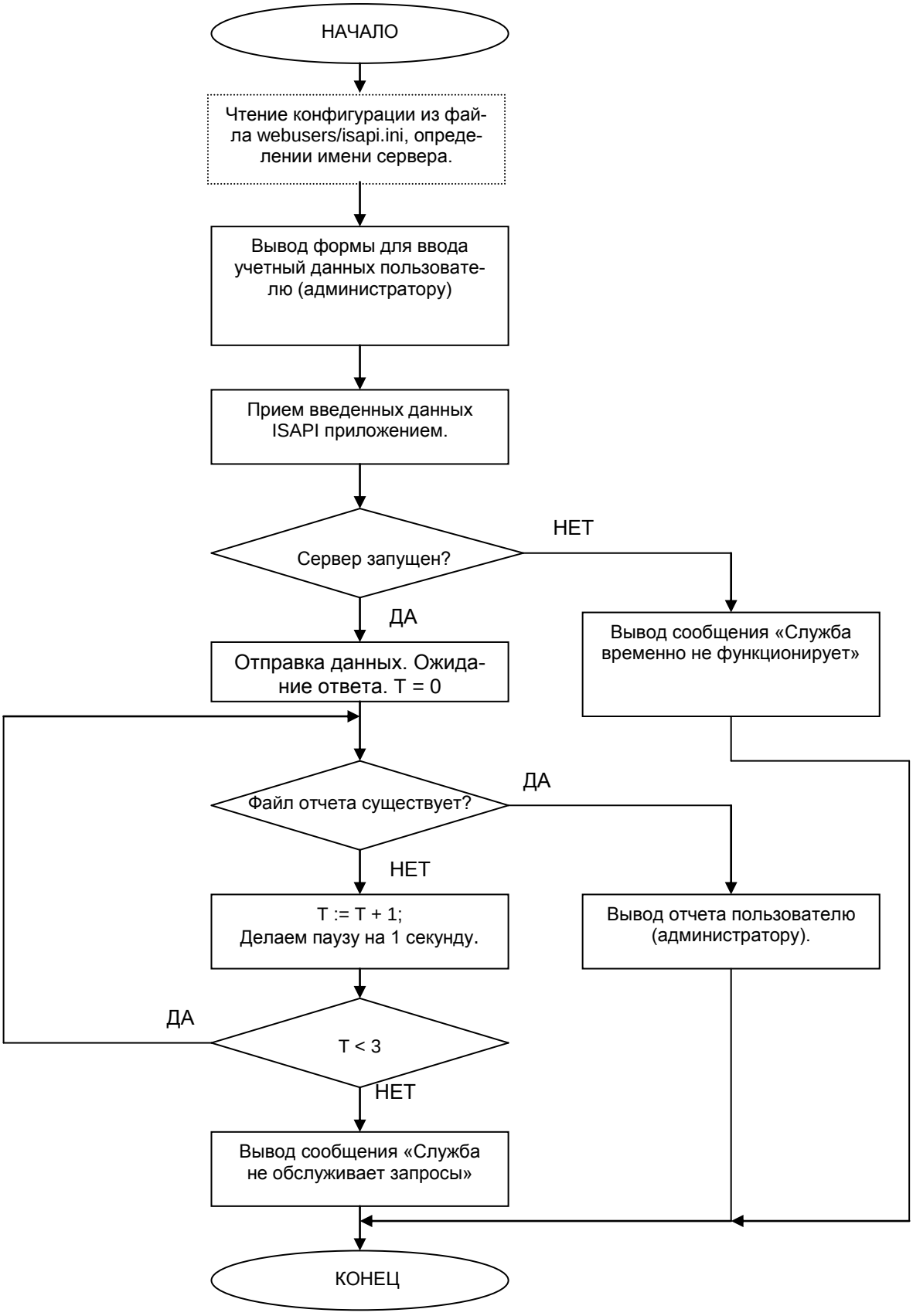


Рис 5. Алгоритм работы ISAPI-приложения (клиента)

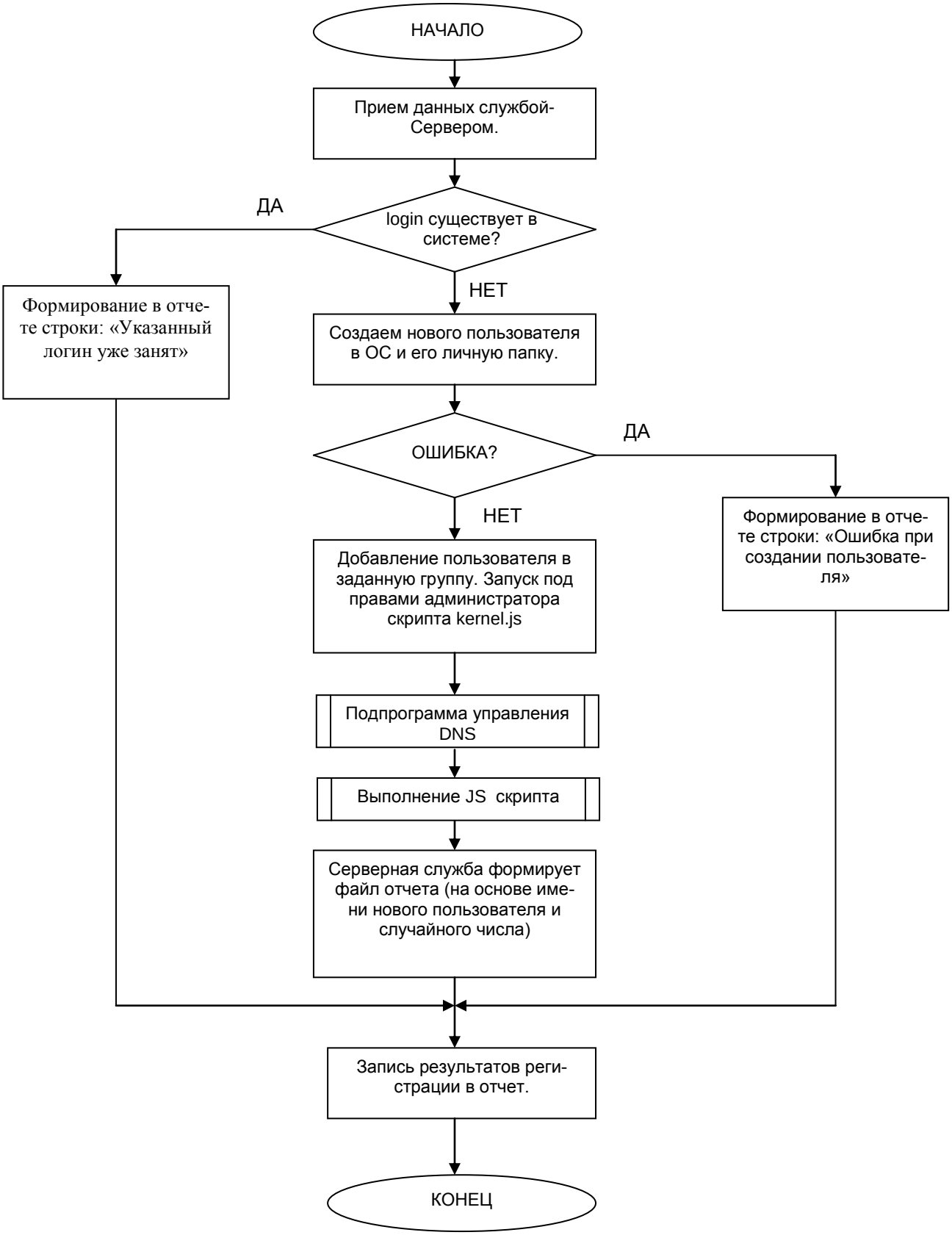
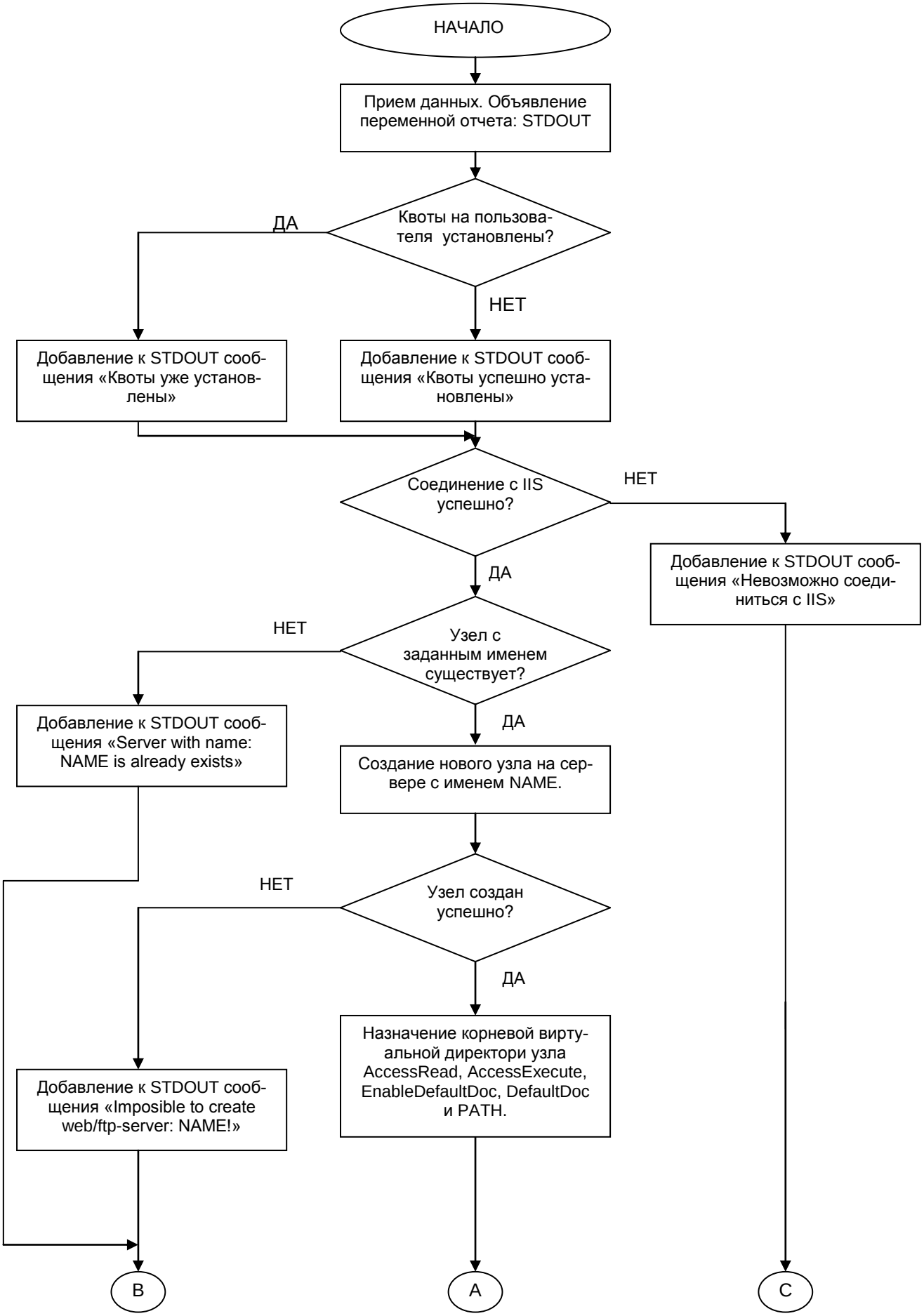


Рис 6. Алгоритм работы сервера


НАЧАЛО
Прием данных. Объявление переменной отчета: STDOUT
Квоты на пользователя установлены?
ДА
НЕТ
Добавление к STDOUT сообщения «Квоты уже установлены»
Добавление к STDOUT сообщения «Квоты успешно установлены»
Соединение с IIS успешно?
НЕТ
ДА
Добавление к STDOUT сообщения «Невозможно соединиться с IIS»
Узел с заданным именем существует?
НЕТ
ДА
Добавление к STDOUT сообщения «Server with name: NAME is already exists»
Создание нового узла на сервере с именем NAME.
Узел создан успешно?
НЕТ
ДА
Назначение корневой виртуальной директори узла AccessRead, AccessExecute, EnableDefaultDoc, DefaultDoc и PATH.
Добавление к STDOUT сообщения «Imposible to create web/ftp-server: NAME!»
B
A
C

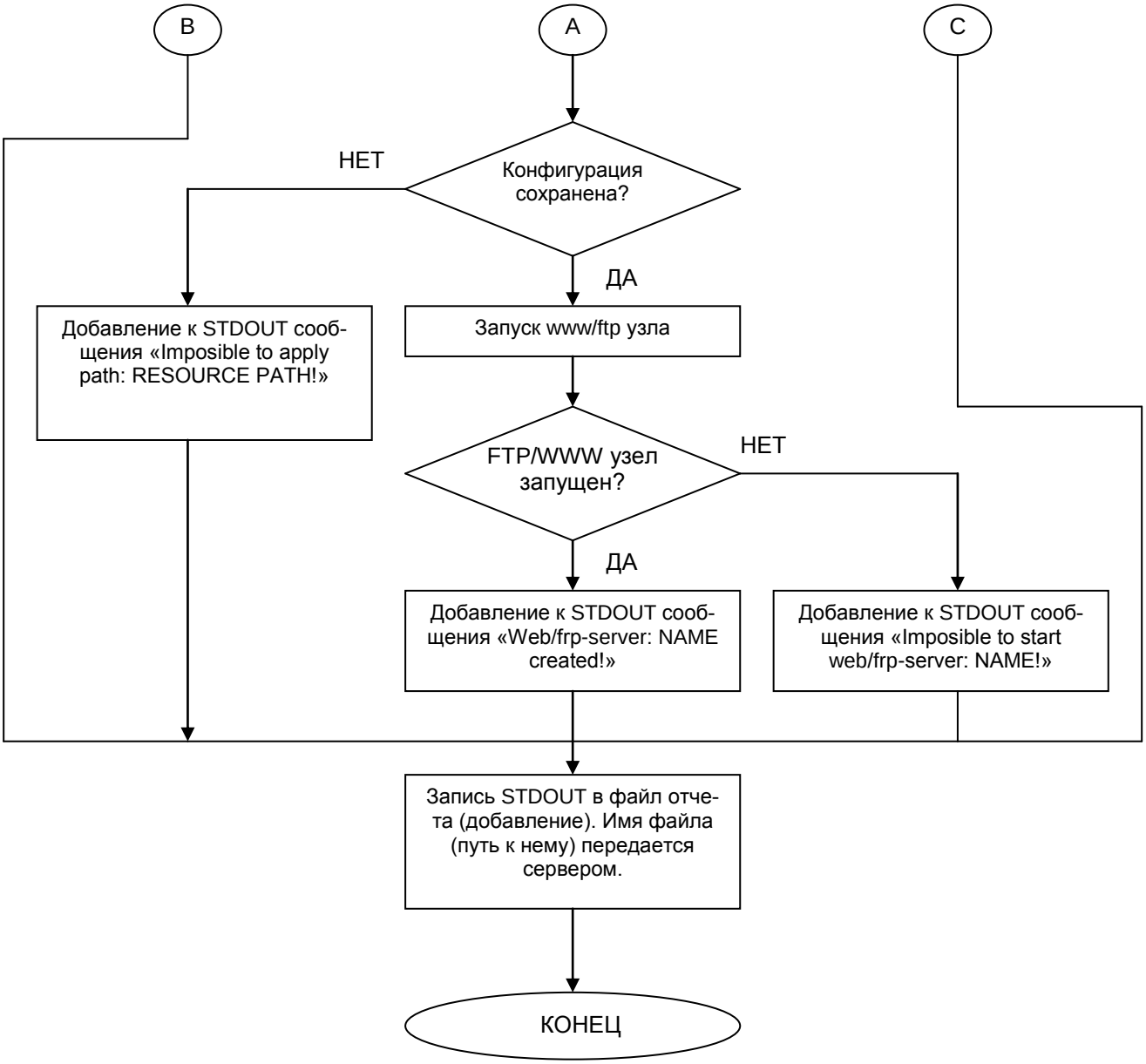


Рис 7. Алгоритм работы скрипта kernel.js

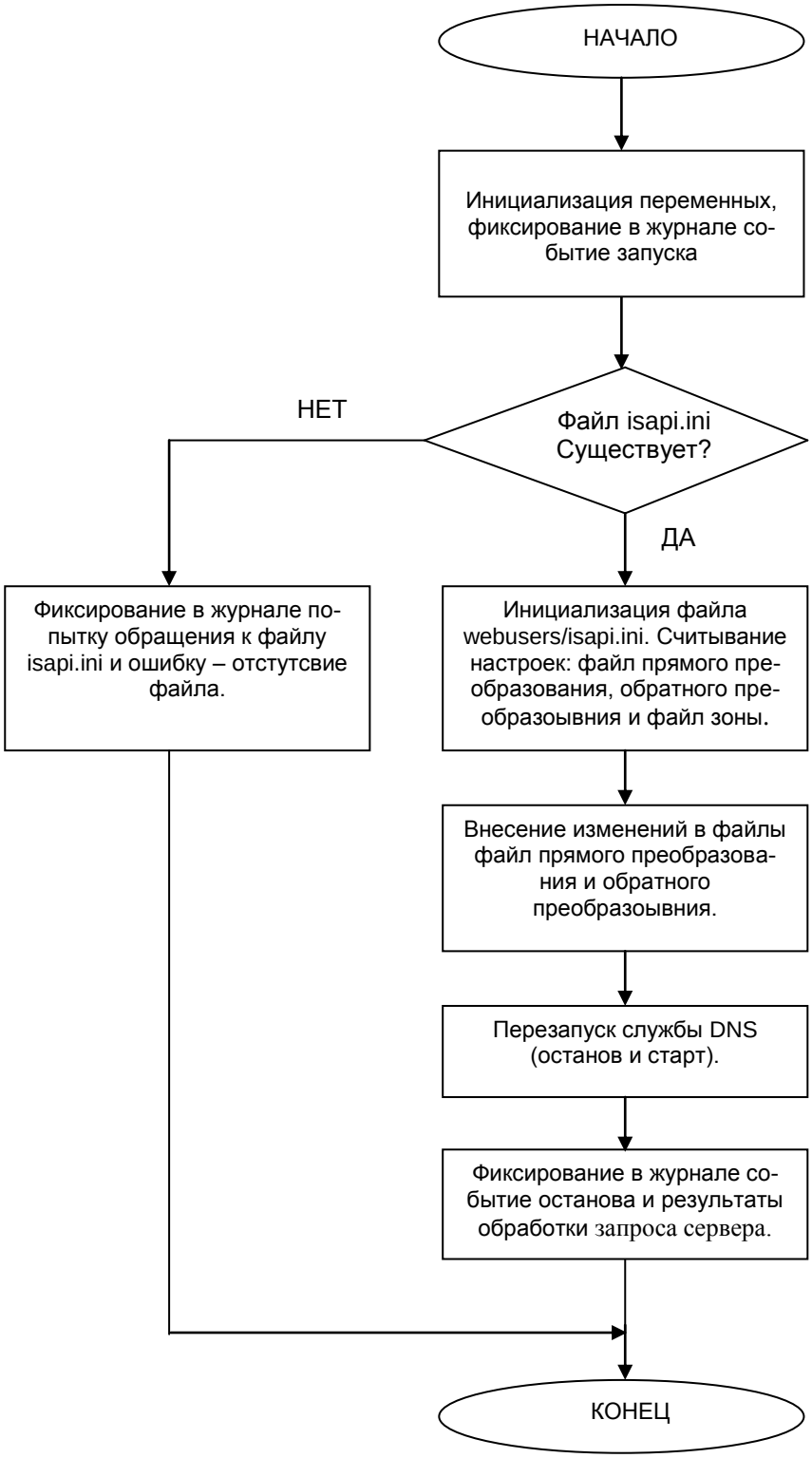


Рис 8. Подпрограмма управления DNS

3.1.4 Проектирование клиентской части

Разработка клиентской части состояла из следующих этапов:

- Разработка непосредственно веб-страницы.
- Реализация взаимодействия веб-страницы с ISAPI приложением.
- Реализация приема данных из формы и их обработка.
- Добавление возможности динамического определения параметра «группа» в зависимости от значения в конфигурационном файле env/group.conf.
- Реализация считывания системного имени сервера.
- Внедрения механизма сокетов в ISAPI приложение.
- Обработка исключительных ситуаций.

Клиентская часть в общих чертах была спроектирована еще в прошлых УИР, тем не менее после более детального тестирования в ней были найдены серьезные недостатки, которое исправлены в текущей версии:

- Некорректная реализация работы с файлом отчетов. В прошлой версии была возможна ситуация, когда, в случае одновременного запроса от двух пользователей с одинаковыми имена, ISAPI мог перепутать файлы отчетов. Из-за этого пользователь 1 мог увидеть пароли пользователя 2. К тому же в системе заводился только один пользователь.
- Ограниченное количество настроек ISAPI из конфигурационного файла.
- Некорректная работа с сокетами, что приводило к зависанию сервиса.

В текущей версии после считывания необходимых конфигурационных данных и данных введенных пользователем, формируется не массив из 5 элементов, как это было ранее, а строка из 5 элементов, которые разделены символом “|” (если пользователей вводит символ “|” в данных при заполнении формы – он будет заменен на “I” [заглавная “i”]). В новой версии элементы строки это: Группа, название веб-узла, logon-имя пользователя, пароль для данного пользователя, имя ожидаемого файла отчета: Logon- случайное_число.log (user-14893756952.log).

3.1.5 Проектирование серверной части

Внешний вид приложения сервера представлен на рисунке 10:

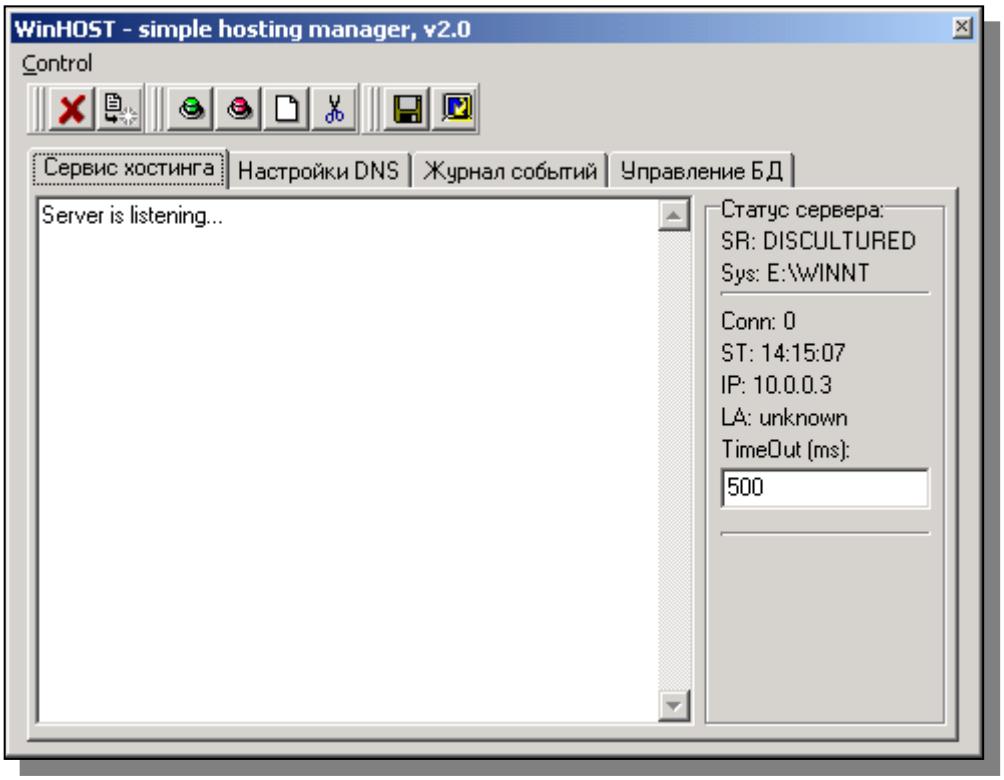


Рис 9. Внешний вид ядра системы.

Разработка серверной части состояла из следующих этапов:

- Внедрения механизма сокетов в сервер-приложение.
- Реализации возможности захвата потока стандартного вывода консольного приложения (cscript.exe)
- Реализация считывания системного имени сервера.
- Реализация последовательного приема и обработки получаемых от клиента данных.
- Реализация возможности программной перезагрузки сервера.
- Внедрение ограничения числа запущенных экземпляров приложения сервера.
- Добавление возможности работы в фоновом режиме и управление сервером из области системного трея.
- Оперирование в рамках сервера с NetAPI функциями – создание и редактирование учетных записей пользователей ОС.

Особенностью функционирования сервера является запуск внешнего скрипта (kernel.js), который квотирует ресурсы для заданного пользователя и работает непосредственно с метабазой IIS, и утилиты управления DNS службой (dnsctrl.exe). Использование внешнего скрипта весьма удобно и эффективно для данной системы.

Управление сервером организовано через иконку в системном трее:

Рис 10. Иконка сервера в системном трее.

Если щелкнуть по выделенной иконке (это и есть иконка сервера), то появится меню:

- Развернуть
- Перезапуск
- Выход

`Развернуть` – выход из фонового режима и активации формы приложения.

`Перезапуск` – перезапуск сервера. Включает в себя полную очистку внутренних переменных окружения, уничтожение и повторное создание сокетов.

`Выход` – закрытие серверного приложения.

Сервер и DNSctrl взаимодействуют с одним конфигурационным файлом. Его формат:

```
[convertions]
froward_conv=w2kserver.net.dns
backward_conv_conv=0.0.10.in-addr.arpa.dns

[zone]
domain_zone=w2kserver.net

[nets]
net=10.0.0
```

Этот файл формируется утилитой DNSctrl. Ниже приведено его краткое описание:

- `[convertions]` – после этого блока перечисляются файлы прямого (forward_conv) и обратного (backward_conv) преобразования адресов DNS.
- `[zone]` – доменная зона, в которой создаются новые узлы
- `[nets]` – адреса сетей, используемых в обратном преобразовании. Это неоходимо для внесения новых значений в файл обратного преобразования.

Для тестирования конфигурационного файла в сервер встроена специальная функция (доступна из меню CONTROL → Test config). На рисунке 12 представлен внешний вид диалогового окна с результатами тестирования:

Рис 11. Диалоговое окно с тестирования конфигурационного файла.

Стоит отметить, что для идентификации сервера в системе использовалась IPC технология – семафоры. Запустившись, сервер создает уникальный семафор, по которому регламентируется повторный запуск сервера, а также идентифицируется наличие сервера в памяти ISAPI приложением.

Механизм сокетов обеспечиват, помимо связи с ISAPI клиентом, последовательную обработку принимаемых данных. Дело в том, что полученный пакет блокирует прием нового, но блокировка не уничтожает пакет, а ставит его в очередь.

При проектировнии сервера, разработчик ориентировался на факт использования данного сервиса под управлением операционной системы MS Windows 2000. Это дало основания снабдить сервер набором специальных NetAPI функции, специфичных для NT технологии. В частности сервер использует:

NetUserAdd – добавление нового пользователя в систему.

NetLocalGroupAdd – добавление новой локальной группы.

NetLocalGroupAddMember – добавление пользователя в заданную группу.

Внешний вид закладки *НАСТРОЙКИ DNS* :

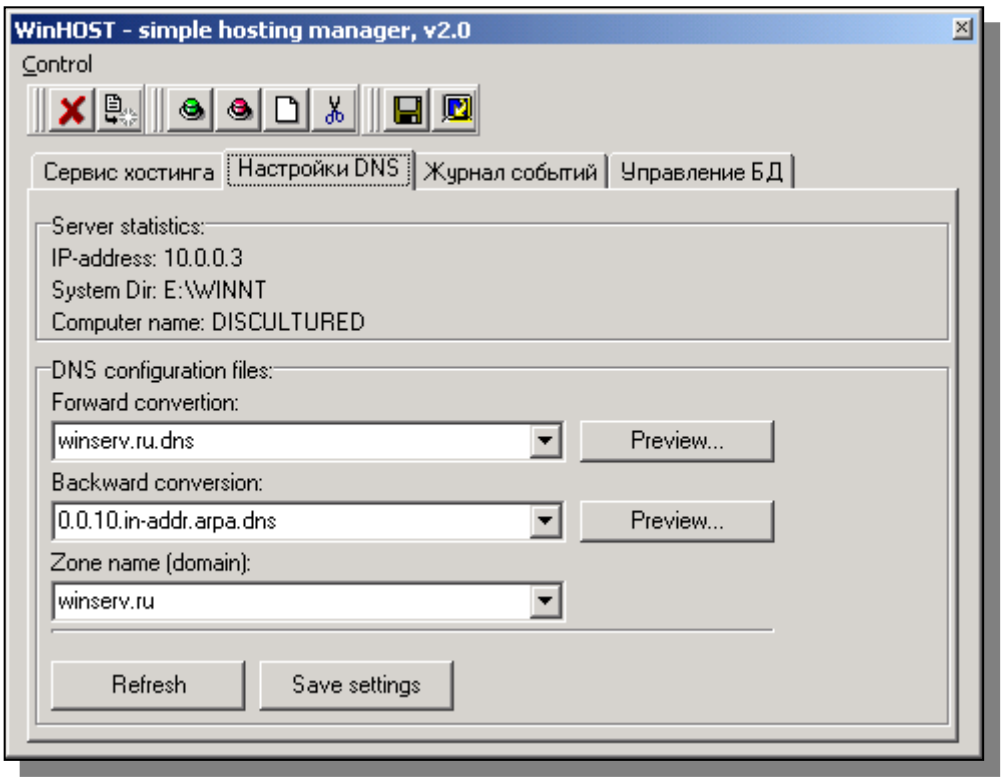


Рис 12. Внешний вид утилиты DNSctrl.

Работа Ядра системы с сервером DNS подразумевает решение задач:

- Работа с системным реестром – получения пути к системной папки.
- Идентификация файлов DNS службы, отвечающих за прямое преобразование, обратное преобразование и доменную зону.
- Реализация возможности их ручного редактирования.
- Реализация возможности конфигурационной настройки из файла.
- Реализация возможности программного тестирования конфигурационного файла.
- Внедрение собственного журнала событий, который фиксирует все операции, выполняемые утилитой.
- Реализация программной перезагрузки DNS службы.

Вид и состав конфигурационного файла описан в предыдущем пункте, отметим, что в Ядре системы предусмотрен собственный движок для тестирования конфигурационного (графическая кнопка *Проверить конфигурацию DNS*). На рисунке представлен внешний вид диалогового окна с результатами тестирования:

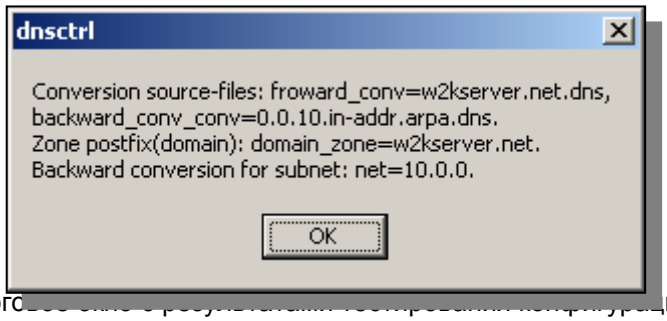


Рис 13. Диалоговое окно с результатами тестирования конфигурационного файла.

Запустившись, Ядро системы сканирует папку с файлами DNS службы и выбирает нужные (файлы обратного и прямого преобразования), помещая их имена в поля **forward conversion** и **backward conversion**, имя зоны формируется на основании имен файлов прямого преобразования.

Администратор должен сам выбрать нужные файлы и зону, которые работоспособны на данном сервере, после этого он должен сохранить настройки в конфигурационный файл, нажав кнопку **Save Settings**. Если IP-адрес сервера, на котором Ядро системы не совпадает с адресом сети файла обратного преобразования, то будет выведено предупреждение:

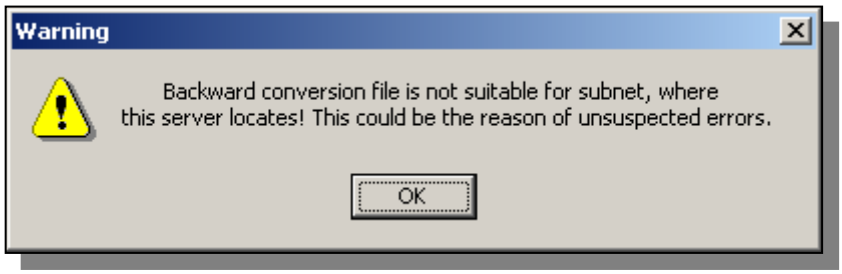


Рис 14. Предупреждение о несовпадении IP-адреса компьютера и адреса сети.

В случае, если конфигурационный файл успешно прочитан, то Ядро системы дополняет файлы DNS службы. В файл прямого преобразования вносится строка:

narkhov CNAME 10.0.0.2

Где narkhov – имя пользователя. В файл обратного преобразования вно-

сится строка:

2 PTR narkhov.w2kserver.net.

Где 2 – адрес компьютера в сети (адрес сети указывается в конфигурационном файле в секции [nets]). Если IP-адрес компьютера 10.0.0.2, а адрес сети 10.0.0, то адрес компьютера в сети = 2. Запись narkhov.w2kserver.net сформирована конкатенацией имени пользователя (narkhov) и доменной зоны (w2kserver.net) через точку – “.”.

После изменения файлов DNS службы производится ее перезапуск. Сначала служба останавливается (на программном уровне вызывается команда **net dns stop**), потом выдерживается пауза 5 секунд и служба запускается (на программном уровне вызывается команда **net dns start**). Пауза необходима для полной уверенности, что служба было остановлена, иначе запуск осуществится до останова, что вызовет ошибку; тем не менее останов все равно завершится, но запуска уже не последует – DNS просто выключится.

3.2 Разработка ПО подсистемы удаленного управления комплексом

Разработка подсистемы удаленного управления комплексом состояла из двух основных этапов. Действительно, удаленное управление ориентированно, в первую очередь, на обычных пользователей, которые должны администрировать собственные ресурсы на сервере – дисковое пространство, личный профиль и свои дополнительные домены. Тем не менее, удаленное управление является одним из основных инструментов работы администраторов системы, которые выполняют схожие с обычными пользователями задачи, но не в масштабах одного пользователя, а системы в целом. Таким образом, первоначально была разработана подсистема удаленного управления пользовательскими ресурсами, а затем подсистема глобального удаленного администрирования.

В техническом задании были поставлены следующие требования к разрабатываемому системе удаленного управления. Они обуславливались, в первую очередь, операционной системой, под управлением которой должен работать сервис, а также стандартным набором функций, присутствующих на аналогичных системах в интернете:

- Система должна быть реализована с помощью технологии ASP;
- Система должна использовать БД MS ACCESS и/или MS SQL Server;
- Система должна управляться через веб-интерфейс. Вход в систему осуществляется на основание уникального логина и пароля. Работа в системе авторизованных пользователей организуется на уровне сессий;
- Глобальная административная часть должна обеспечивать следующие функциональные возможности: создание/модификация/удаление пользователей системы и создание новых хостов в системе;
- Пользовательская часть системы должна реализовывать следующие функции: полное управление заданной областью (т.е. каталогом) дискового пространства сервера;
- В пользовательской части системы должна быть предусмотрена операция добавления новых файлов (закачки) на сервер.

3.2.1 Сценарии администрирования сайтов пользователей (пользовательская часть)

Сценарий – как в пользовательской, так и в административной части – (web-приложение, включающее в себя 6 отдельных модулей), управляющий пользовательскими ресурсами (дисковым пространством и БД), функционирует по следующему алгоритму:

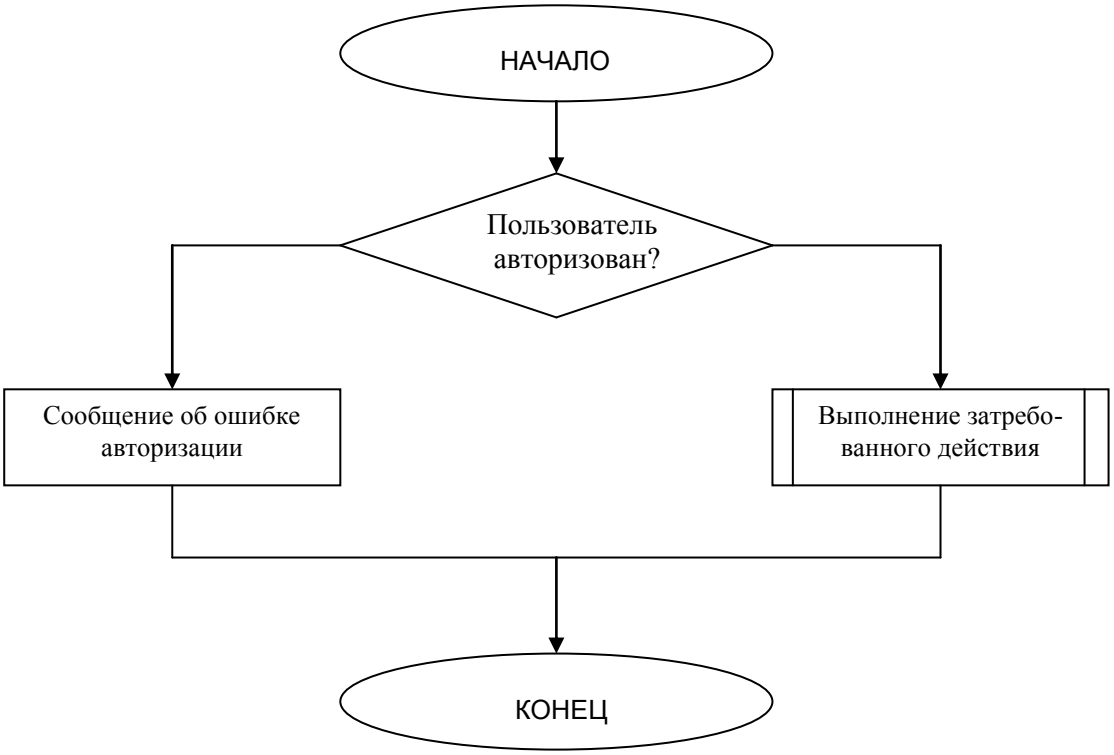


Рис 15. Общий алгоритм работы административных сценариев.

Список выполняемых действий в пользовательской части:

Действия	**Описание**
login	Вывод страницы авторизации
auth	Проверка введенных учетных данных и вывод главной страницы пользовательской части (отображение содержимого – файлы/каталоги – корневой папки пользователя).
chngdir	Смена папки и вывод содержимого (файлы/каталоги) нового каталога
exit	Выход из пользовательской части.
newdir	Вывод страницы для создание новой директории

createdir	Непосредственно создание новой директории в текущей папке (операция над файловой системой).
rename	Вывод страницы для переименования выбранных файлов/каталогов.
renamefs	Непосредственно переименования выбранных файлов/каталогов (операция над файловой системой)..
showlog	Вывод на экран содержания файлов с расширением .txt, .log.
upload	Добавление новых файлов в текущий каталог.

Внешний вид пользовательской части представлен на рис. 16

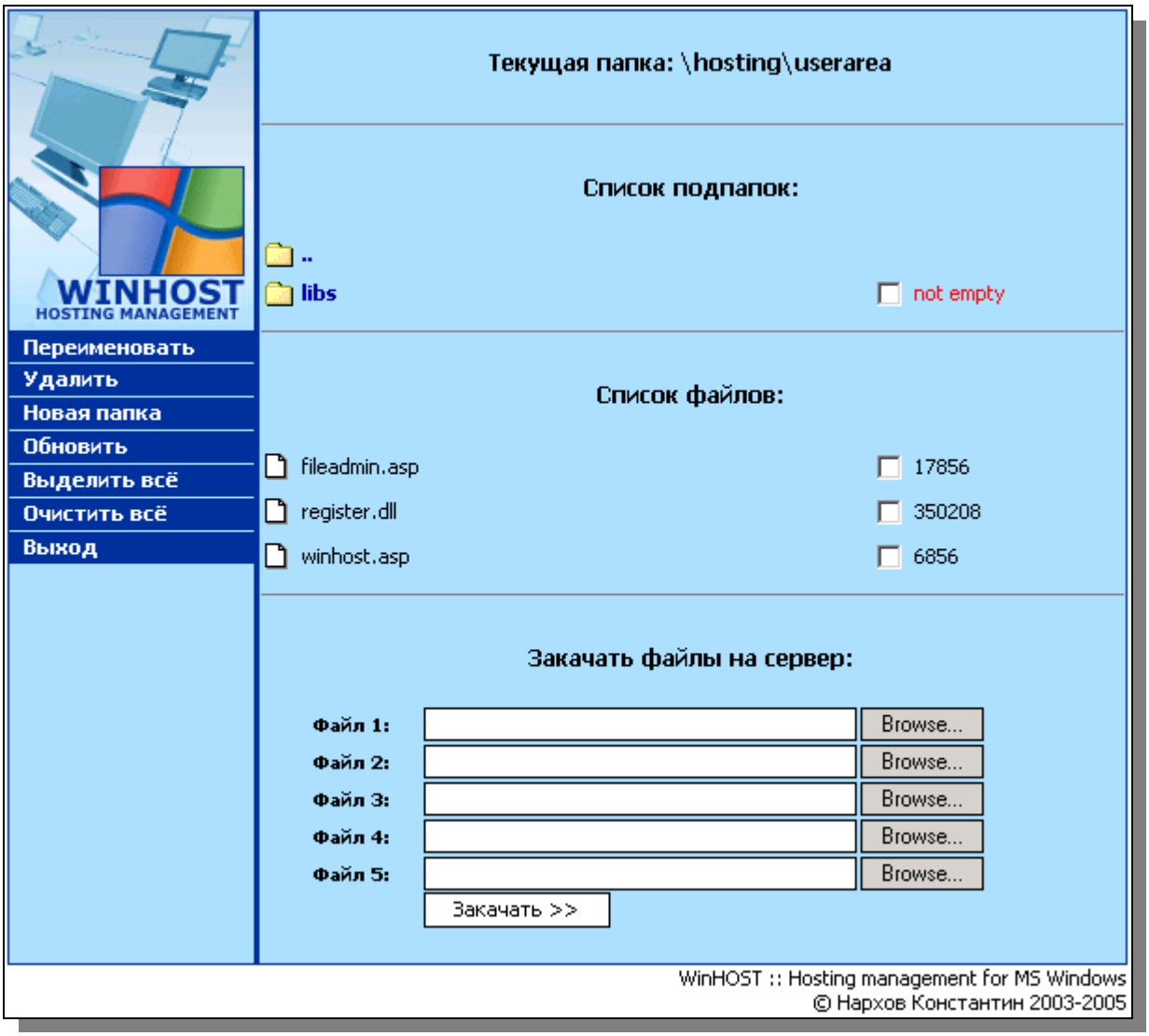


Рис 16. Пользовательская часть.

При авторизации пользователь попадает в личную директорию (т.е. закрепленный за данным пользователем каталог на сервере). Смещение относительно этой пап-

ки отображается в информационном поле Текущая папка. Под смещением понимается глубина входа в подкаталоги личной директории. Далее располагаются списки подкаталогов текущей папки и размещенных в ней файлов – Список подпапок и Список файлов соответственно. Внизу страницы размещена форма добавления новых файлов на сервер (максимально 5 штук за раз): Закачать файлы на сервер. Рядом (справа) с каждым объектом списков помещено поле выделения – CHECKBOX, – с помощью которого можно пометить необходимую папку или файл для дальнейших операций. За данным полей располагается флаг «пустоты» для каталогов или размер в байтах для файлов.

В левой части страницы размещена навигация по пользовательской части:

1. Переименовать – операция переименования выделенных объектов из списков. Основана на вызове функции объекта Scripting.FileSystemObject GetFile или GetFolder и изменении свойства NAME выбранного файла или каталога.
2. Удалить – операция удаления выделенных объектов из списков. Основана на вызове функции объекта Scripting.FileSystemObject GetFile или GetFolder и последующего метода Delete выбранного файла или каталога с флагом true (удаление непустых каталогов или файлов, предназначенных только для чтения).
3. Новая папка - операция создания нового каталога в текущем. Основана на вызове функции объекта Scripting.FileSystemObject GetFolder, SubFolders и метода Add с названием нового каталога в качестве параметра.
4. Обновить – обновляет текущую страницу (иногда необходимо, чтобы увидеть произведенные изменения)
5. Выделить всё – выделяет все объекты в списках каталогов и файлов.
6. Очистить всё – снимает флаг выделения со всех объектов в списках каталогов и файлов.
7. Выход – выход из пользовательской части, завершение сессии.

3.2.2 Сценарии администрирования подсистемы в целом (административная часть)

Список выполняемых действий в административной части:

Действия	Описание
goindex	Вывод главной страницы административной части
adduser	Вывод страницы для добавление нового пользователя
addusrdb	Непосредственно добавление нового пользователя (операция над БД)
deluser	Удаление пользователя (операция над БД)
moduser	Вывод страницы для изменения регистрационных данных пользователя
regdomain	Регистрация нового хоста в системе
modusrdb	Непосредственно изменения регистрационных данных пользователя (операция над БД)

Внешний вид административной части представлен на рис. 17:

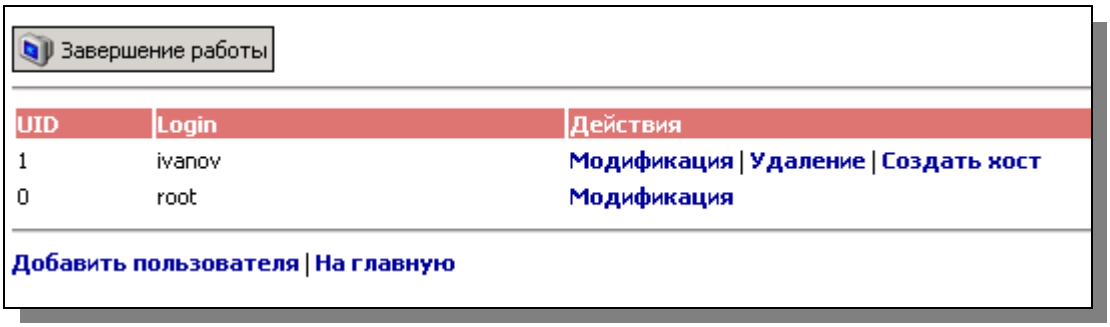


Рис 17. Административная часть.

В верхней части страницы размещена кнопка `Завершение работы` – выход из административной части с закрытием сессии. Далее представлен список пользователей, зарегистрированных в системе. `UID` – учетный идентификатор пользователя в системе, `Login` – псевдоним пользователя, `Действия` – список операций над пользователем. Внизу страницы расположены ссылки для добавления нового пользователя в систему (`Добавить пользователя`) и перехода на главную страницу административной части (`На главную`).

МОДИФИКАЦИЯ пользователя предусматривает изменения данных о персоне (фамилии, имени, отчества, домашнего адрес и проч.), а также статуса бло-

кировки и пути к домашнему каталогу. Форма модификации личных данных пользователя представлена на рис. 18:

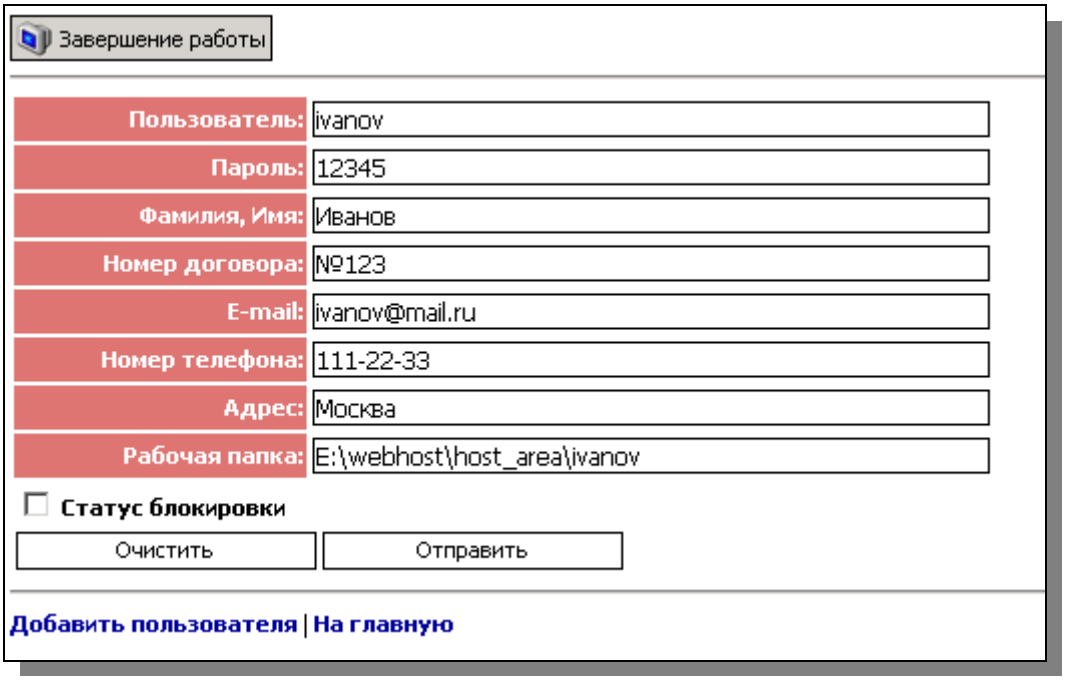


Рис 18. Форма модификации данных.

Ссылка *УДАЛЕНИЕ* реализует удаление выбранного пользователя из системы. Следует уточнить, что удаление осуществляется на уровне базы данных (уничтожаются все записи ассоциированные с данным пользователем) и файловой системы (удаляется все содержимое из домашнего каталога и помещается индексный файл с сообщением пользователь удален из системы). Хост и записи в конфигурационных файлах DNS сервера сохраняются.

Ссылка *СОЗДАТЬ ХОСТ* создает узел в MS IIS сервере и запись в DNS сервере системы. При успешном выполнении операции выводится сообщение:

```
Params report. Group=hostingusers, Site=pavlov, User=pavlov,
Passw=pavelpavlov, Log=959446367.log.
Registering user. User is created!
Creating INDEX.HTML ... done!
User is added to group!
USER URL: pavlov.winserv.ru
Creating DNS account ... done!
Creating WWW... DONE!
```

Данный отчет содержит следующую информацию о результатах создания нового WWW-узла в системе:

- `Params report` – строка параметров, которые будут переданы административному скрипту и на основе которых будет создана новая запись в DNS, каталог пользователя и лог-файл;
- `Registering user` – отображения статуса создания пользователя;
- `Creating INDEX.HTML` – отображения статуса создания индексного файла в домашнем каталоге пользователя;
- `Creating DNS account` – отображение статуса создания записей в конфигурационных файлах DNS сервера;
- `Creating WWW` – отображение статуса создания WWW-узла в IIS сервере.

Копия данного отчета сохраняется в каталоге `%sys_dir%\webhost\host_area\hosting\log`, где `%sys_dir%` - диск, на котором установлена операционная система. Файл отчета: `%Login%_959446367.log`, где `%Login%` – логин-псевдоним созданного пользователя.

При выполнении операции *СОЗДАТЬ ХОСТ* возможны следующие сообщения об ошибках:

- `Service is blocked!` – невозможно идентифицировать Ядро системы. Программа `regserver.exe` не запущена или «зависила».
- `Registration failed. Call administrator for details!` – некорректная структура каталогов в системе. Скрипт, создающий новый хост, не может найти файл отчета Ядра системы. Убедитесь, что каталог `%sys_dir%\webhost\host_area\hosting\log`, где `%sys_dir%` - диск, на котором установлена операционная система, существует. Также проверьте, что исполняемый файл Ядра системы расположен в каталоге `%sys_dir%\webhost\`.
- `Disturb of service! Wait while server will be restarted.` – система ожидает результата создания предыдущего WWW-узла. Повторите попытку создания через несколько минут.

4. Реализация системы

4.1 Выбор СУБД для БД системы

Прежде всего, необходимо определиться с операционной системой, на которой будет возможно использовать систему. Поскольку наиболее вероятно, что сервер баз данных для системы и Web-сервер будут находиться на одной машине, этот выбор повлияет на последующий выбор ОС для СУБД и самой СУБД.

В данной работе выбор фактически однозначно пал на операционную систему семейства Microsoft Windows NT 2000 Server. Это может быть как Windows 2000 Server, так и Windows 2000 Advanced Server.

Причин к тому было несколько:

1. Для данной системы существует свой Интернет-сервер Microsoft Internet Information Server 5.0 (IIS), который обладает всеми необходимыми для запуска Web-приложений возможностями;
2. Данная ОС достаточно проста в администрировании;
3. Достаточно безопасная ОС
4. У разработчика, так и у заказчика разрабатываемой системы есть в наличии лицензионный пакет с ОС Microsoft Windows 2000 Server.

После выбора операционной системы выбор СУБД сузился до двух: Microsoft SQL-Server 2000 и Microsoft Access 2000. Разработаны варианты реализации системы как для первой, так и для второй.

Выбор этих СУБД связан с:

1. Специальным требованием к системе стало требование быстрой установки, а MS Access изначально встроена в ОС и поэтому ее использование не требует установки никакого дополнительного ПО;
2. MS Access поддерживает запросы написанные на языке SQL;
3. Фирма Microsoft рекомендует использование MS Access для организации динамических сайтов;
4. Использование данной MS SQL Server значительно упрощает задачу разработчика, т.к. она поддерживает запуск хранимых процедур на сервере, позволяющих упростить все операции включения – изменения данных.

4.2 Реализация БД системы

Для реализации поставленных перед системой задач, требовалось организовать базу данных для хранения информации о пользователях системы. Кроме этого, для аудита работы с системой сделана специально таблица протоколирования действий. Данные, которые хранятся в ней, не используются системой и необходимы лишь для статистического использования.

Таблицы базы данных, содержат следующие поля:

Табл. 2. Описание полей таблиц БД

Протоколирование действий (Logs)	**Авторизация (Auth)**
1. Уникальный LogsID 2. UID или AID 3. IP-адрес с которым работает UID или AID 4. Время события 5. Описание события	1. Уникальный AuthID 2. UID или AID 3. IP, с которым работает UID или AID 4. Временный пароль 5. Время входа в систему
Пользователи (Users)	**Домены (UserSysData)**
1. Уникальный UID. 2. Логин 3. Пароль 4. ФИО 5. Номер договора 6. E-mail 7. Телефон 8. Адрес 9. Дисковая квота 10. Дата регистрации 11. Статус	1. Уникальный DomID 2. UID пользователя 3. Наименование 4. IP, к которому привязан 5. Рабочая папка 6. Дата регистрации

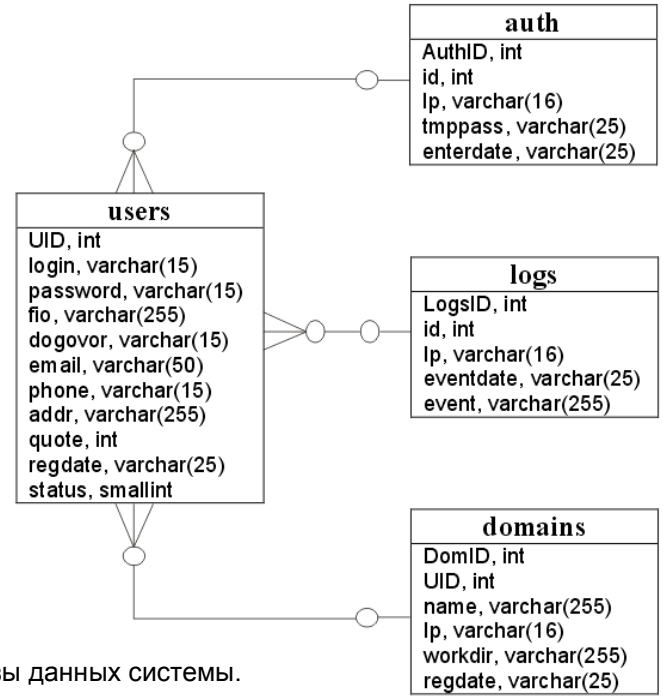


Рис 18. Схема базы данных системы.

Поскольку было принято решение разработать два варианта реализации системы, было создано, две базы данных: для MS Access и для MS SQL-Server. БД для MS Access создавалась с помощью одноименной программы MS Access 2000 (см. Рис. 19), которая входит в состав пакета Microsoft Office 2000.

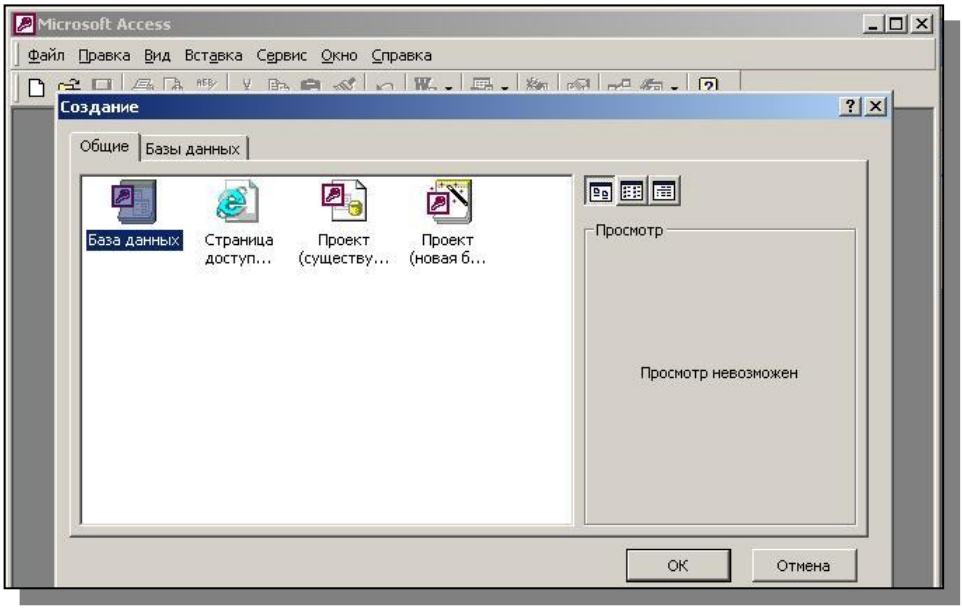


Рис. 19. Создание БД, используя MS Access 2000

Поскольку замечено, что при большом количестве отношений внутри БД данная СУБД достаточно нестабильно работает даже с небольшим количеством пользователей, все межтабличные связи были реализованы методами Web-приложения, а не СУБД, что правда, вызвало для разработчика необходимость самому следить за сохранением целостности данных.

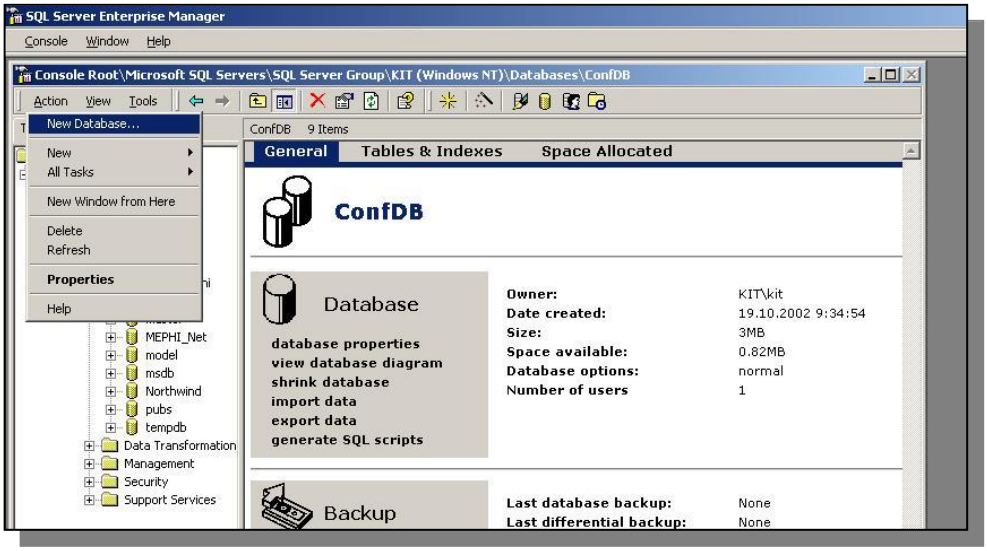


Рис. 20. Создание БД при помощи SQL-Server Enterprise Manager

4.3 Реализация ПО системы

Для реализации Ядра системы была использована среда Delphi 6.0. Этот выбор был сделан по следующим причинам:

- Простота настройки и использования Delphi;
- Поддержка средой Delphi 6.0 системных функций Windows 2000;
- Поддержка средой Delphi 6.0 сетевых функций WinSock 2.0;
- Наличие у автора опыта работы с данной средой разработки;
- Среда Delphi постоянно совершенствуется и появляются новые компоненты для использования в ее составе.

ISAPI-приложение также разрабатывалось в среде Delphi по причинам, изложенным выше. Кроме того, Delphi компилирует ISAPI библиотеки полностью совместимые с MS ISS 5.0.

Для разработки данного Web-приложения автор воспользовался технологией Active Server Pages (см. 1.2.2.1) по следующим причинам:

- Данная технология разработана все той же фирмой Microsoft специально для создания динамических сайтов под управлением Интернет-сервера IIS и предоставляет разработчику большинство необходимых функций;
- В данной технологии сценарии выполняются в рамках единого процесса IISa, в отличие от CGI и ISAPI – приложений, для запуска которых создаются отдельные процессы. Этот факт увеличивает скорость работы динамического сайта;
- Как указывалось ранее (см. Введение), ранее на кафедре уже были наработки по близкой тематике, и, разумеется, разработчику хотелось каким-то образом их использовать. В данных наработках использовалась технология ASP, что тоже повлияло на данный выбор;
- И, наконец, последняя причина заключается в том, что эта технология не «тупиковая», и постоянно совершенствуется (создана технология ASP.NET, в которой поддерживается идеология ASP).

В качестве языка разработки серверных сценариев использовался язык JScript (Java Script).

В технологии ASP основным механизмом доступа к базам данных является технология ADO (ActiveX Data Objects). Именно эта технология и была использована в данной работе. Рассмотрим ее немного более подробно:

Одним из основных объектов, доступных из серверного скрипта, является объект Server, именно этот объект используется для создания экземпляров таких объектов ActiveX, как, например, ADODB.RecordSet.

Объект Connection (подсоединение) создается методом CreateObject объекта Server, и ссылка на него помещается в переменную. Когда объект создан, его можно использовать для открытия подсоединения либо к любому источнику данных ODBC, либо для подсоединения непосредственно к БД, используя OLEDB-драйвер соответствующей СУБД. Следующий фрагмент кода устанавливает подсоединение к БД MS Access используя соответствующий OLEDB-драйвер:

```
<%
' Объявляем переменную
Dim objConnection
' Создаем объект Connection
Set objConnection = Server.CreateObject("ADODB.Connection")
' Открываем подсоединение к источнику данных
    objConnection.Open "Provid-
er=Microsoft.Jet.OLEDB.4.0;Data Source=" & rootdir &
"\DB\my_db.mdb"
%>
```

Здесь objConnection — переменная для объектной ссылки на экземпляр объекта Connection. Метод Open устанавливает подсоединение, принимая в качестве аргументов имя источника данных, идентификатор пользователя и пароль. Когда подсоединение установлено, получать информацию из источника данных можно при помощи объекта Recordset (набор записей). Этот объект может выполнять оператор SELECT языка SQL и возвращать набор записей, удовлетворяющих этому запросу. Как и объект Connection, Recordset создается методом CreateObject.

Разработка административного скрипта kernel.js проводилась с помощью технологии Windows Script Host. Данная технология выбрана по следующим причинам из того, что WSH предъявляет минимальные требования к объему оперативной памяти и является очень удобным инструментом для автоматизации повседневных задач пользователей и администраторов операционной системы Windows. Используя сценарии WSH, можно непосредственно работать с файловой

системой компьютера, а таже управлять работой других приложений (серверов автоматизации). При этом возможности сценариев ограничены только средствами, которые предоставляют доступные серверы автоматизации. Перечислим только наиболее очевидные задачи, для автоматизации которых прекрасно подходят сценарии WSH:

- Организация резервного копирования на сетевой сервер файлов с ло кальной машины, которые отбираются по какому-либо критерию;
- Быстрое изменение конфигурации рабочего стола Windows в зависимости от задач, выполняемых пользователем;
- Автоматический запуск программ Microsoft Office, создание там сложных составных документов, распечатка этих документов и закрытие приложений;
- Управление работой приложений, не являющихся серверами автоматизации, с помощью посылки в эти приложения нажатий клавиш;
- Подключение и отключение сетевых ресурсов (дисков и принтеров). П Создание сложных сценариев регистрации для пользователей;
- Выполнение задач администрирования локальной сети (например, добавление или удаление пользователей).

4.4 Создание установочного пакета системы

Поскольку специальным требованием к системе было требование легкой и быстрой ее установки на сервер, было принято решение создать установочный пакет системы.

Напомним, что для успешной установки и первоначальной настройки необходимо, чтобы на сервере был установлен MS IIS 5.0/6.0 сервер и DNS сервер, сконфигурированный посредством файлов.

Система реализована таким образом, что для ее успешного функционирования достаточно выполнить четыре основных шага:

- Подготовить каталог к копированию файлов – если каталог с таким же именем существует – удалить его.
- Скопировать файлы системы на серверный компьютер;
- Установка на скопированные файлы атрибутов – чтение/запись и права доступа для ВСЕХ пользователей;
- Создание корректного файла соединения с базой данной для web-приложения. Имеется ввиду, что в файле прописан корректный путь к базе данных с учетом места, куда скопированы файлы.
- Вывод сообщения об успешном копировании;
- Запуск Ядра системы – `regserver.exe`.

После запуска Ядра системы, администратор должен произвести первоначальную настройку системы, которая состоит из следующих этапов:

- Конфигурация DNS (закладка *Настройка DNS*);
- Создание корневых WWW- и FTP-узлов системы;

После выполненный операций система готова к экспуатации.

4. Тестирование и отладка

В процессе тестирования и отладки сменилась реализация взаимодействия клиента и сервера, реализованную в данном комплексе в прошлой УИР. Кроме этого, в системе был выявлен ряд неисправностей, которые, в целом, удалось исправить. Следует упомянуть то обстоятельство, что текущая версия комплекса - 2.0, а за весь отладочный процесс версия сервиса изменялась 7 раз (первоначальный номер 1 – после прошлой УИР).

Условия тестирования.

Тестовый стенд имел следующую конфигурацию: процессор Intel Pentium III 1 ГГц, материнская плата Asus CUSL2-C, 256 Mb ОЗУ, видеокарта ATI Rage 128, HDD Maxtor 2R015H1 объемом 40 Гб. Операционная система Microsoft Windows 2000 Server, с установленными ServicePack 4, IIS 5.0, WHS 5.0 и Internet Explorer 6.0. В качестве среды разработки использовался продукт Borland Delphi 5.0

Подготовительный этап.

На подготовительном этапе необходимо создать узел, на котором будет располагаться ISAPI клиент нашего хостинга. Пусть IP-адрес сервера 10.0.0.1, а имя доменной зоны w2kserver.net, тогда создадим узел hosting.w2kserver.net. Для этого перейдем в *Мой компьютер/панель управления/администрирование/Диспетчер служб Интернета.* Далее раскроем подсписок у корневого элемента списка. В нем мы можем наблюдать имеющиеся на вашем сервере WWW, FTP и MAIL службы. Выделите корневой элемент списка и кликнете правую кнопку мыши. В появившемся меню выберете пункт СОЗДАТЬ → виртуальный WWW-узел. Следуя подсказкам мастера создание завершите операцию. Также нужно переконфигурировать DNS службу для того, чтобы узел hosting.w2kserver.net был доступен по DNS имени (т.е. прямо по hosting.w2kserver.net). Для этого перейдите на *Мой компьютер/панель управления/администрирование/DNS.* Далее создайте в зоне (предположим, что она существует. Если нет, то создайте зону) w2kserver.net для прямого преобразования узел hosting с адресом 10.0.0.2. В поле *Создать соответствующую PTR-запись* установите флаг – обратное преобразование обновится автоматически. Перезагрузите DNS службу. Создайте в системной папке Windows (обычно

WINNT\System32) каталог webusers и поместите в него пустой файл isapi.ini (формат файла см. в пункте 3.6).

Непосредственно тестирование.

Запускаем сервер (каталог, откуда запускаем не важен – важно чтобы он содержал подкаталог scripts со скриптом kernel.js и подкаталог dns с файлами DNSctr(dnsctrl.exe,dnsconf.ini и events.log). Создаем папке сервера (сервера хостинга – regserver.exe) host_area сервера каталог admin и admin\env, копируем в них содержимое ISAPI клиента с CD, не нарушая общей древовидной структуры. Теперь зайдем в утилиту управления WWW-Ftp узлами (см. Подготовительный этап). Создадим на веб-узле hosting.w2kserver.net виртуальный каталог admin. Для этого на выбранном узле нажмите правую кнопку мыши, далее *Создать/Виртуальный каталог*. Укажите в свойствах каталога возможность запуска скриптов/приложений и путь к физическому (пака_сервера_хостинга\host_area\admin) и. После проделанных операции ISAPI клиент станет доступен по адресу http://hosting.w2kserver.net/admin/.Закроем сервер и проверим как ведет себя клиент в отсутствии сервера. Нажмем в ISAPI кнопку *«Submit»* (по адресу http://hosting.w2kserver.net/admin/register.dll). Результат: сообщение *«Service is blocked».* Запустим сервер вновь. Корректно заполним поля формы клиента (попробуем создать нового пользователя, который явно отсутствует в системе), нажмем *«Submit».* Через долю секунды клиент выведет лог о работе:

```
Params report. Group=hostingusers, Site=NEW_USER, User=extirpator,
Passw=konstantin, Log=85254537.log.
Registering user. User is created!
Group is created!
Creating INDEX.HTML ... done!
User is added to group!
USER URL: extirpator.w2kserver.net
Creating DNS account ... done!

Creating WWW... DONE!
Creating FTP... DONE!
```

Первая строка лога **(Params report)** указывает на то, что после нее перечислены переданные скрипту параметры регистрации **(Group:, Site:, User:, Passw:, Log:).** На их основе создавались дисковые квоты, WWW/FTP-узлы и новая учетная запись в ОС. Далее мы видим надпись о том, что пользователь, группа и стартовый HTML файл в физической папке пользователя созданы. После этого система информирует нас об успешном включении пользователя в созданную группу. Показывается полный адрес WWW/FTP-узлов пользователя и выводится информация об успешном переконфигурировании DNS службы. Далее мы можем увидеть информацию непосредственно о создании WWW/FTP узлов. Данный тест показывает работоспособность прямого пути алгоритма.

Проверим функциональность системы в том случае, если пользователь в системе не существует, а виртуальный каталог уже имеет место быть. Дело в том, что при удалении какого-либо пользователя из системы администратор может производить операцию удаления только учетной записи. Удаление виртуального каталога не нужно, но желательно. Если виртуальный каталог не будет удален, все файлы удаленного пользователя из учетных записей системы будут доступны из вне. Тем не менее, при повторном создании такого пользователя сервисом каталог будет очищен.

Пример. Пусть пользователь extirpator не существует в системе, но в IIS существуют узлы http:// и ftp:// extirpator.w2kserver.net. Повторно создадим newuser пользователя хостинга:

```
Params report. Group=hostingusers, Site=NEW_USER, User=extirpator,
Passw=konstantin, Log=1000596233.log.
Registering user. User is created!
Group is created!
Creating INDEX.HTML ... done!
User is added to group!
USER URL: extirpator.w2kserver.net
Creating DNS account ... done!

Creating WWW... Server with name: extirpator.w2kserver.net is already exists!
```

Система сообщила нам, что пользователь, группа и стартовый HTML файл в физической папке пользователя созданы. После этого система информирует нас об успешном включении пользователя в созданную группу. Показывается полный адрес WWW/FTP-узлов пользователя и выводится информация об успешном переконфигурировании DNS службы. Также мы можем увидеть, что узлы с указанным именем созданы не были. Следует упомянуть о том, что в процессе тестирования сервиса версии 1.0 никаких зависаний и отказов в работе не было зафиксировано.

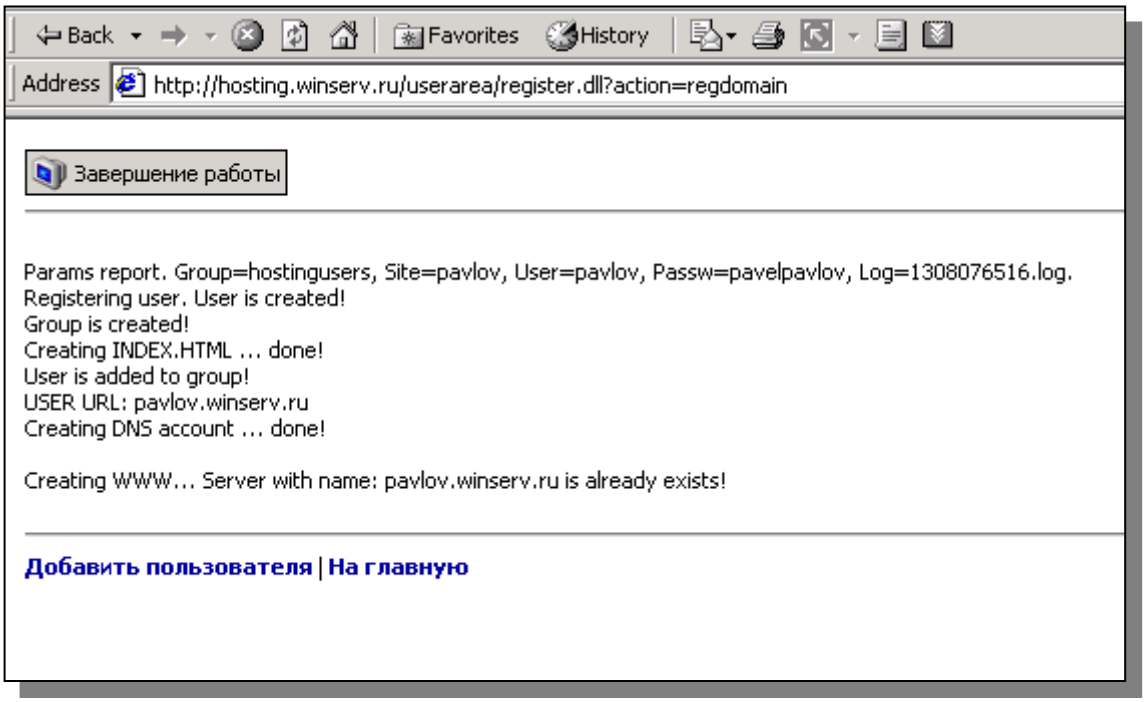


Рис. 21. Результат тестирования ПО.

4.2 Исключительные ситуации в работе комплекса

В процессе разработки возникали ошибки и неправильные варианты решения. Перечислим некоторые из них:

- Разработка связи клиент-сервер на основе OLE/DCOM технологии.

Ошибка: запуск COM-объекта осуществляется в контексте ISAPI приложения. Это недопустимо.

Решение: Отказ от использования данной технологии.

- Запуск скрипта kernel.js из ISAPI под правами администратора. Тестировались

API функции CreatePoccessAsUser и CreateProcessWithLogonW.

Ошибка: Для запуска CreatePoccessAsUser необходима првиления System (ISAPI ее конечно же не имеет). Запустить CreateProcessWithLogonW неудалось по до сих пор неустановленым причинам (скорее всего это связано с контекстом запуска – функция запускалась в рамках CGI процесса, который существенно отличается от обычных процессов).

Решение: Отказ от использования указанных функций.

- Обработка в имени веб-узла пробелов.

Ошибка: В случае работы с веб-узлом содержащим в своем имени пробелы (например, Default Web Site) в kernel.js передавалась только части имени (т.е. Default).

Решение: Дело в том, что формат запуска kernel.js следующий: cscript.exe scripts/kernel.js param1 param2 param3 param4. При использовании в имени узла пробелов (имя узла – это param1). Строка вызова выглядела: cscript.exe scripts/kernel.js Default Web Site param2 param3 param4, т.е. в скрипт передавались только Default Web Site и param2 (каждое слово в param1 интерпертировалось как отдельный параметр).

Модификация вызова на cscript.exe scripts/kernel.js “param1” “param2” “param3” “param4” решила данную проблему.

Остальные ошибки, появляющиеся в процессе разработки, не имеют интереса для рассмотрения. В общем, они довольно просты и имеют скорее статус опечаток.

4.1 Анализ результатов тестирования

Техническое задание предъявляло к данной системе определенные требования (см. 3.1.1). Проанализировав результаты тестирования можно смело сказать, что разработанный сервис полностью удовлетворяет поставленному ТЗ.

А именно:

- Имеет полный инструментарий для работы с учетными данными ОС. Способен создавать новых пользователей, новые локальные группы, назначать членство пользователей в разных группах;
- Имеет возможность назначать и удалять дисковые квоты на создаваемых пользователей;
- Полная поддержка DNS службы через конфигурационные файлы.
- Сервис настраивает как WWW-, так и FTP-доступ к файлам пользователя;
- Тестирование показало, что сервис устойчив к иключительным ситуациям. А часть, работающая непосредственно с пользователем – ISAPI, наиболее защищена и устойчива к сбоям. Таким образом пользователь никогда не получит сообщения, типа “Error 500” или проч;
- Сервис способен управлять локальным IIS сервером: создавать и переопределять свойства виртуальных каталогов, создавать WWW/FTP узлы.
- Полный цикл работы сервиса длится чуть более 1 секунды;
- Использование расширенной технологии исключений позволяет перехватывать внутри сервера почти все ошибки и обрабатывать их;
- Спроектированы систему удаленного администрирования дискового пространства сервера и используемой в работе базы данных;
- Сервис прост и удобен в работе.

Заметим, что служба данного сервиса сожет быть установлена на ОС Windows 2000 Professional и Windows 2000 Server.

Заключение

В результате дипломного проектирования была создана универсальная система для эффективного размещения и дальнейшего управления данными в сети Internet. Данная система представляет собой серверную часть (скомпилированная программа для управления системными ресурсами) и динамический сайт (web-приложение), отвечающий за удаленное управление. Были разработаны базы данных системы под СУБД Microsoft Access и Microsoft SQL-Server 7/2000, а так же средства создания их программным путем. Были разработаны программные модули web-приложения, которые представляют собой серверные сценарии пользователя, администратора и рецензента. Данные модули реализованы на технологии активных страниц ASP, работающей под управлением серверов Microsoft Windows 2000/2003 Server или Microsoft Windows NT 4 Server. В качестве языка разработки серверных скриптов использовался язык JScript. Для разработки скриптов, работающих на стороне клиента, использовался язык JavaScript 1.3, так как он поддерживается большинством популярных браузеров. В составе серверной части используются административные скрипты, необходимые для управления MS IIS сервером и квотирования дискового пространства пользователей. Для данного набора скриптов использовалась технология Windows Script Host. В качестве языка программирования использовался язык JScript.

Был создан инсталляционный пакет системы, позволяющий быстро установить ее на сервер даже человеку, не обладающему навыками администрирования Microsoft Internet Information Server. При создании установочного пакета также использовалась технология Windows Script Host.

При разработке системы для определения функций, доступных пользователю и администратору проводился сравнительный анализ сайтов, использующих web-приложение для удаленного администрирования серверов: «Boom.ru», «Newmail.ru», и «Narod.ru», велась переписка с Web-мастерами и администраторами таких известных интернет-провайдеров, как «Мастерхост» и «Агава». Система состоит из 76 файлов и занимает порядка 1-го мегабайта дискового пространства сервера.

Основными преимуществами разработки являются простота установки и настройки системы и расширенная подсистема администрирования, которая предоставляет удаленное управление дисковым пространством сервера и базой данных системы. Кроме того, web-приложение, входящее в состав системы, обла-

дает широким функциональных, что делает его возможным для использования в качестве основного средства удаленного управления дисковым пространством на любом NT-сервере.

Одним из важнейших недостатков системы является зависимость от операционной системы. Это и зависимость сервера – система может быть использована только на сервере с установленной операционной системой Microsoft Windows NT/2000/2003 Server. Тем не менее, задача дипломного проекта связывалась с разработкой системы управления ресурсами именно NT-сервера. Это обусловлено развитием Windows-серверов в последнее время и дефицит подобных программных продуктов для них.

В процессе работы автор получил навыки разработки программных комплексов с применением современных Internet-технологий, разработки служебных средств с помощью технологии Windows Script Host, а так же установки и администрирования СУБД Microsoft SQL Server 7/2000.

После завершения тестирования, система была внедрена в практическое использование московским интеренет-провайдером «М35 Нетворкс».

Список литерутуры

1. Ногл М. TCP/IP. Иллюстрированный учебник: Пер. с англ. – М.: ДМК Пресс, 2001.
2. Попов А.В. Windows Script Host для Windows 2000/XP. – СПб.: БХВ-Петербург, 2003.
3. Дарахвелидзе П.Г., Марков Е.П. Программирование в Delphi 4. – СПб.: БХВ-Петербург, 1999.
4. Microsoft Windows 2000 server. Учебный курс MCSA/MCSE: Пер. с англ. – 3-е изд. – М.: Издательско-торговый дом «Русская Редакция», 2002.
5. Материалы электронной документации к серверу MS IIS 5.0

Приложение 1. Инструкция по эксплуатации

Установка системы.

Установка системы состоит из следующих этапов:

1. Вставьте в дисковод чтения компакт дисков дистрибутив системы.
2. Запустите установочный файл `install.js`.
3. Убедитесь, что в стандартную установку Вашей Windows входят MS IIS 5.0/6.0 и DNS сервер. Для этого войдите в *Панель Управления* и выберите подпункт *Администрирование*. Далее перейдите в подраздел *Службы* и найдите в списке сервисов Службу веб-публикаций (World Wide Web Publishing Service) и DNS Сервер (DNS Server). Данные службы должны быть запущены. В случае отсутствия указанных сервисов в списке установите их.
4. Убедитесь, что DNS Сервер настроен на работу через конфигурационные файлы (т.е. не связан с Active Directory).
5. Убедитесь, что в MS IIS 5.0/6.0 нет ни одного FTP-узла. Если таковые присутствуют, то отключите их.

Первоначальное конфигурирование.

Первоначальное конфигурирование состоит из следующих действий:

1. Перейдите в каталог `%sys_dir%\webhost`, где `%sys_dir%` - диск, на котором установлена операционная система. Далее запустите файл ядра системы: `regserver.exe`. Появится окно приложения, на котором присутствуют следующие элементы управления:
   - Меню *CONTROL* – глобальное управление программой. Подменю *CLEAR TERMINAL* – очистка статусного окна сервера (закладка *СЕРВИС ХОСТИНГА*). Подменю *RESTART SERVICE* – «горячая» перезагрузка программы. Подменю *ABOUT* – справка, подменю *EXIT* – завершение программы.
   - Панель графических кнопок, выполняющие действия:
      - завершение программы;
      - минимизация окна программы;
      - запуск DNS службы;
      - останов DNS службы;
      - проверка конфигурации DNS службы;

 - очистка журнала событий;

 - сохранение изменений в базу данных;

 - импорт данных из базы данных;

- Набор закладок для изменения и отображения информации о состоянии сервера, базе данных, журнала событий и сервере DNS.

2. Перейдите на закладку *НАСТРОЙКИ DNS* и выберите нужные значения для полей *forward conversion, backward conversion, zone name.* Для корректного заполнения этих полей нужно знать IP адрес сервера, адрес подсети, в которой находится сервер, и доменное имя сервера.
3. После заполнения полей из п. 2 нажмите кнопку *SAVE SETTINGS* для сохранения изменений.
4. Прейдите на закладку *УПРАВЛЕНИЕ БД*, из списка пользователей выберите *ROOT* и нажмите правую кнопку мышки. В контекстном меню выберите пункт *СОЗДАТЬ ХОСТ*.
5. Зайдите в оснастку MS IIS. Для созданного в п. 4 узла `http://hosting.доменное имя сервера`, разрешите в *СВОЙСТВАХ* запуск скриптов и скомпилированных приложений (*Executables & Scripts*), а также добавьте в список документов по умолчанию (*Default Documents*) файлы `index.html` и `index.htm`.
6. Для FTP-службы сервера добавьте в список *SECURITY ACCOUNTS* группу `hostingusers`.

Проверка работоспособности.

Для проверки работоспособности системы после первоначальной настройки необходимо открыть окно браузера и перейти на узел `http://hosting.доменное имя сервера`. Если вы увидите перед собой страницу – настройка произведена успешно.

Базовое администрирование системы.

Администрирование системы подразделяется на два типа: удаленное администрирование и администрирование на стороне сервера. Первое подразумевает управление пользователями, узлами и дисковым пространством через web-интерфейс с удаленного компьютера. Администрирование на стороне сервера

предполагает использования программы `regserver.exe` для управления нужными ресурсами сервера. Для удаленного управления прейдите на `http://hosting.доменное имя сервера`. Для администрирования базы данных и узлов используйте ссылку *ВХОД ДЛЯ АДМИНИСТРАТОРОВ*, для управления дисковым пространством – *УПРАВЛЕНИЕ ФАЙЛАМИ САЙТА*. Логин пароль соответственно после первоначальной настройки: `root/root`.

Устранение неполадок.

Система может работать с неполадками из-за следующих причин:

1. Некорректно сконфигурированы службы DNS и/или MS IIS.

   Решение: убедитесь, что служба DNS работает и настроена на управление через файла, также проверьте работоспособность MS IIS.

2. Установлены неправильные права доступа на каталог `%sys_dir%\webhost`, где `%sys_dir%` - диск, на котором установлена операционная система.

   Решение: убедитесь, что на *ВСЕ* пользователи имею права на чтение и запись в нем.

3. Программа `regserver.exe` «зависла».

   Решение: откройте утилиту «Диспетчер задач» («Task Manager»). В списке процессов найдите `regserver.exe` и завершите его.

4. Не работает FTP-служба.

   Решение: убедитесь, что на сервере работает только один узел FTP: `ftp://hosting.доменное имя сервера`. Для этого сначала остановите все FTP-узлы сервера, а затем запустите нужный.

5. Невозможно администрировать удаленно файловую систему - не найдена база данных.

   Решение: зайдите в каталог `%sys_dir%\webhost\host_area\hosting\userarea\libs`, где `%sys_dir%` - диск, на котором установлена операционная система. Откройте файл `datastore.txt` и проверьте путь к базе данных. Внимание! В качестве разделителя между каталогами и файлами в записи пути должен быть ‘\\’!!!