ОГЛАВЛЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1
«СРЕДСТВА ОПЕРАЦИОННОЙ СРЕДЫ WINDOWS ДЛЯ НАСТРОЙКИ СЕТЕВОГО ОБОРУДОВАНИЯ»

Цель работы: изучить основные, стандартные средства современных операционных сред для настройки параметров сетевого оборудования.

Компьютерная сеть (вычислительная сеть, сеть передачи данных) – система связи компьютеров и/или компьютерного оборудования (серверы, маршрутизаторы и другое оборудование). Для передачи информации могут быть использованы различные физические явления, как правило – различные виды электрических сигналов, световых сигналов или электромагнитного излучения.

По назначению компьютерные сети распределяются:

1. вычислительные;
2. информационные;
3. смешанные.

Вычислительные сети предназначены главным образом для решения заданий пользователей с обменом данными между их абонентами.

Информационные сети ориентированы в основном на предоставление информационных услуг пользователям.

Смешанные сети совмещают функции первых двух.

Для классификации компьютерных сетей используются разные признаки, выбор которых заключается в том, чтобы выделить из существующего многообразия такие, которые позволили бы обеспечить данной классификационной схеме такие обязательные качества:

1. возможность классификации всех, как существующих, так и перспективных, компьютерных сетей;
2. дифференциацию существенно разных сетей;
3. однозначность классификации любой компьютерной сети;
4. наглядность, простоту и практическую целесообразность классификационной схемы.

Определенное несоответствие этих требований делает задание по выбору рациональной схемы классификации компьютерной сети достаточно сложной, такой, которая не нашла до этого времени однозначного решения. В основном компьютерные сети классифицируют по признакам структурной и функциональной организации.

По территориальной распространенности компьютерные сети подразделяются на:

PAN (Personal Area Network) – персональная сеть, предназначенная для взаимодействия различных устройств, принадлежащих одному владельцу.

LAN (Local Area Network) – локальные сети, имеющие замкнутую инфраструктуру до выхода на поставщиков услуг. Термин «LAN» может описывать и маленькую офисную сеть, и сеть уровня большого завода,

занимающего несколько сотен гектаров. Зарубежные источники дают даже близкую оценку – около шести миль (10 км) в радиусе. Локальные сети являются сетями закрытого типа, доступ к ним разрешен только ограниченному кругу пользователей, для которых работа в такой сети непосредственно связана с их профессиональной деятельностью.

CAN (Campus Area Network – кампусная сеть) – объединяет локальные сети близко расположенных зданий.

MAN (Metropolitan Area Network) – городские сети между учреждениями в пределах одного или нескольких городов, связывающие много локальных вычислительных сетей.

WAN (Wide Area Network) – глобальная сеть, покрывающая большие географические регионы, включающие в себя как локальные сети, так и прочие телекоммуникационные сети и устройства. Пример WAN – сети с коммутацией пакетов (Frame relay), через которую могут «разговаривать» между собой различные компьютерные сети. Глобальные сети являются открытыми и ориентированы на обслуживание любых пользователей.

Термин «корпоративная сеть» также используется в литературе для обозначения объединения нескольких сетей, каждая из которых может быть построена на различных технических, программных и информационных принципах.

По типу функционального взаимодействия:

1. Клиент-сервер;
2. Смешанная сеть;
3. Одноранговая сеть;
4. Многоранговые сети.

Задание: настроить сетевое оборудование операционной системы для работы в локальной сети с возможностью доступа в Интернет. Привести скриншоты процесса настройки.

Контрольные вопросы:

1. Что такое локальная сеть?
2. Что такое протокол?
3. Назначение IP-адреса
4. На какие типы делятся сети по функциональному взаимодействию?
5. Поясните аббревиатуру WAN
6. Поясните аббревиатуру LAN

ЛАБОРАТОРНАЯ РАБОТА № 2
«СРЕДСТВА ОПЕРАЦИОННОЙ СРЕДЫ WINDOWS ДЛЯ АНАЛИЗА СОСТОЯНИЯ СЕТИ»

Цель работы: изучить основные, стандартные средства современных операционных сред для исследования состояния сети и сетевых соединений.

Концепция вычислительных сетей является логическим результатом эволюции компьютерных технологий. В настоящее время вычислительные сети продолжают развиваться, причем достаточно быстро. Разрыв между локальными и глобальными сетями при этом постоянно сокращается во, многом из-за появления высокоскоростных территориальных каналов связи, не уступающих по качеству кабельным системам локальных сетей. В глобальных сетях появляются службы доступа к ресурсам, такие же фундаментальные, как и службы локальных сетей. Подобные примеры в большом количестве демонстрирует глобальная сеть – Internet, базирующаяся на протоколе TCP/IP. Протокол TCP/IP использует сочетание нескольких схем адресации. Самый нижний уровень адресации задается сетевыми аппаратными средствами. На следующем, более высоком уровне используется Internet-адресация (которую чаще всего называют IP-адресацией).

Существует несколько пакетов прикладных программ, которые используются при отладке сети на уровне TCP/IP, которые, в большинстве своем, дают низкоуровневую информацию.

Простейшим из таких средств является команда ping. Она служит для принудительного вызова ответа конкретной машины. Формат команды следующий:

ping [-t] [-a] [-n число] [-l размер] [-f] [-i TTL] [-v TOS] [-r число] [-s число] [[-j списокУзлов] | [-k списокУзлов]] [-w интервал] адрес машины

где:

-t – Отправка пакетов на указанный узел до команды прерывания;
-a – Определение адресов по именам узлов;
-n число – Число отправляемых запросов;
-l размер – Размер буфера отправки;
-f – Установка флага, запрещающего фрагментацию пакета;
-i TTL – Задание времени жизни пакета (поле "Time To Live");
-v TOS – Задание типа службы (поле "Type Of Service");
-r число – Запись маршрута для указанного числа переходов;
-s число – Штамп времени для указанного числа переходов;
-j списокУзлов – Свободный выбор маршрута по списку узлов;
-k списокУзлов – Жесткий выбор маршрута по списку узлов;
-w интервал – Интервал ожидания каждого ответа в миллисекундах.

Команда ping позволяет проверить функционирование основных элементов сети. В выполнении команды ping участвуют система маршрутизации, схемы разрешения адресов и сетевые шлюзы, поэтому для достижения успешного результата сеть должна быть в рабочем состоянии.

Другим средством анализа работоспособности сети является программа tracert. Эта программа позволяет выявлять последовательность шлюзов, через которую проходит IP-пакет на пути к пункту своего назначения. Выходной информацией команды является простой список машин, начиная с первого шлюза и заканчивая пунктом назначения. Синтаксис программы следующий:

tracert [-d] [-h максЧисло] [-j списокУзлов] [-w интервал]
имя_машины

где:
-d – Без определения адресов по именам узлов;
-h максЧисло – Максимальное число переходов при поиске узла;
-j списокУзлов – Свободный выбор маршрута по списку узлов;
-w интервал – Интервал ожидания каждого ответа в миллисекундах.

Еще одним средством изучения состояния сети является команда netstat. Она предназначена для отображения статистики протокола и текущих сетевых подключений TCP/IP. Синтаксис команды следующий:

netstat [-a] [-e] [-n] [-s] [-p имя] [-r] [интервал]

где:
-a – Отображение всех подключений и ожидающих портов; (Подключения со стороны сервера обычно не отображаются).
-e – Отображение статистики Ethernet. Этот ключ может применяться вместе с ключом -s;
-n – Отображение адресов и номеров портов в числовом формате;
-p имя – Отображение подключений для протокола "имя": tcp или udp. Используется вместе с ключом -s для отображения статистики по протоколам. Допустимые значения "имя": tcp, udp или ip;
-r – Отображение содержимого таблицы маршрутов;
-s – Отображение статистики по протоколам. По умолчанию выводятся данные для TCP, UDP и IP. Ключ -p позволяет указать подмножество выводящихся данных.
интервал – Повторный вывод статистических данных через указанный интервал в секундах. Для прекращения вывода данных нажмите клавиши CTRL+C. Если параметр не задан, сведения о текущей конфигурации выводятся один раз.

Варианты заданий.

Выполнить анализ прохождения IP-пакетов к адресам, заданным преподавателем, с помощью команд ping, tracert, netstat.

Контрольные вопросы:

1. Назначение команды ping?
2. Что такое IP-адрес?
3. Назначение таблицы маршрутизации?
4. Основные функции команды netstat.
5. Назначение программы tracert.

ЛАБОРАТОРНАЯ РАБОТА №3.
СРЕДСТВА ПРОЕКТИРОВАНИЯ И МОДЕЛИРОВАНИЯ СЕТЕЙ.
«ПОСТРОЕНИЕ ОДНОУРОВНЕВОГО СЕТЕВОГО ПРОЕКТА И ОЦЕНКА ЕГО РАБОТЫ»

Цель работы: приобрести практические навыки в построении одноуровневого сетевого проекта и оценке его работы с помощь системы моделирования NetCracker.

Построить гипотетическую сеть в которой рабочие станции объединяются с помощью коммутатора, придерживаясь следующих ограничений:

1. использовать стандартные рабочие станции (LAN workstations / Ethernet workstations);
2. использовать стандартный коммутатор (Switches/Switch);
3. тип соединения с коммутатором – Ethernet;
4. тип протокола при выборе трафика – Small Office;
5. мониторингу подлежат любые соединения.

NetCracker – система представляет собой CASE-средства автоматизированного проектирования, моделирования и анализа компьютерных сетей. Позволяет провести эксперименты, результаты которых могут быть использованы для обоснования выбора типа сети, сред передачи, сетевых компонент оборудования и программно-математического обеспечения.

Программные средства NetCracker позволяют выполнить сбор соответствующих данных о существующей сети без останова ее работы, создать проект этой сети и выполнить необходимые эксперименты для определения предельных характеристик, возможности расширения, изменения топологии и модификации сетевого оборудования с целью дальнейшего ее совершенствования и развития.

С помощью NetCracker можно проектировать компьютерные сети различного масштаба и назначения: от локальных сетей, насчитывающих несколько десятков компьютеров, до межгосударственных глобальных сетей, построенных с использованием спутниковой связи. В составе программного обеспечения NetCracker имеется мощная база данных сетевых устройств ведущих производителей: рабочих станций, серверов, сред передачи, сетевых адаптеров, повторителей, мостов, коммутаторов, маршрутизаторов, используемых для различных типов сетей и сетевых технологий.

NetCracker позволяет разрабатывать многоуровневые проекты с заданной проектировщиком степенью детализации; при этом имеется достаточно удобный интерфейс и средства быстрого просмотра всех уровней проекта. Для реализаций функций имитационного моделирования в составе NetCracker предусмотрены средства задания характеристик трафиков различных протоколов; средства визуального контроля заданных параметров; средства накопления статистической информации и формирования отчетной документации о проведенных экспериментах.

Варианты заданий.

N mod 4	Количество рабочих станций
0	4
1	6
2	8
3	10

N mod 2	Тип соединения станций
0	попарно
1	через одну

N mod 2	Тип трафика между станциями
0	однонаправленный
1	двунаправленный

N mod 2	Тип мониторинга соединения
0	Текущее использование соединения (Current Utilization) в процентном отношении и в виде прямоугольной диаграммы
1	Текущая рабочая нагрузка (Current Workload) в единицах байт/сек

N mod 3	Количество соединений, подлежащих мониторингу
0	2
1	3
2	4

ЛАБОРАТОРНАЯ РАБОТА № 4
ПРИМЕНЕНИЕ JAVASCRIPT ПРИ СОЗДАНИИ WEB СТРАНИЦ.

Цель работы: изучить основные возможности языка JavaScript для создания динамических Web-страниц.

Язык JavaScript (иногда его называют Mocha — по аналогии с Java, а иногда — язык сценариев) — это язык программирования, который позволяет встраивать выполняемое содержимое в документы, написанные на языке HTML. В сущности, это усеченный язык программирования, который представляет собой более свободную интерпретацию языка Java, хотя и менее сложен в сравнении с последним. С появлением JavaScript была ликвидирована пропасть, разделявшая создание текстов на языке HTML и программирование на языке Java. JavaScript позволяет разрабатывать выполняемое содержимое, не вдаваясь в тонкости сложного языка программирования.

Программы на языке JavaScript являются автономными и помещаются в документы, написанные на языке HTML. Программа на языке JavaScript интерпретируется самим броузером при загрузке документа, в который помещен ее код. В этом и состоит одно из основных отличий программ на языке JavaScript от апплетов языка Java, хранящихся отдельно от документа HTML, к которому они относятся.

Программы, написанные на языке JavaScript, способны решать самые различные задачи и могут быть настолько сложными (или простыми), насколько это требуется. Когда-то фраза "программирование на HTML" вызвала целую волну критики среди программистов, работающих в сети Internet. Но теперь документ HTML может действительно содержать значительную "программируемую" часть.

Сравнительная характеристика JavaScript и Java

В основном Java и JavaScript различаются количеством возможностей и сложностью. JavaScript имеет более свободную в сравнении с Java реализацию, поэтому в нем, например, можно не объявлять переменные, а преобразование типов данных осуществляется значительно проще. К тому же, исходный код программы на языке JavaScript не нужно компилировать, в отличие от Java-апплетов: язык JavaScript интерпретируемый. Интерпретатор JavaScript читает программу строка за строкой и сообщает об ошибках (если таковые есть) после каждой прочитанной строки (а не после обработки всего текста программы, как это делает компилятор языка Java). Исходный код программы на языке Java должен быть преобразован в формат с побайтовым представлением до выполнения программы, и, следовательно, программы на языке JavaScript можно разрабатывать и отлаживать значительно быстрее.

Как и язык Java, JavaScript является объектно-ориентированным языком, хотя в нем нет ни классов, ни встроенных механизмов наследования, которые являются стандартными для Java. В программах на языке Java используются классы объектов, и эти программы являются полностью

объектно-ориентированными. И Java, и JavaScript относятся к числу безопасных языков – в том смысле, что в каждом из них реализована поддержка средств, которые не допускают записи на жесткий диск данных, полученных из неизвестного источника.

Для чего можно использовать JavaScript?

Все события, которые генерируются браузером, такие как нажатия кнопок, обработка полей и перемещение между страницами, можно перехватить и обработать средствами JavaScript.

Язык JavaScript обеспечивает, помимо средств обработки отдельных обращений пользователя к гиперсвязям, возможность распознавания момента перехода на другую страницу и выполнения соответствующих действий при наступлении этого события. Язык JavaScript прекрасно подходит для решения рутинных ежедневных задач, таких как проверка достоверности данных, обработка форм, а также для выполнения действий над строковыми и числовыми значениями, т.е. тех задач, которые нельзя решить с помощью существующих средств языка HTML. С его помощью можно динамически создавать документы HTML, то есть такие документы, которые создаются программой на языке JavaScript, а не самим пользователем. Следовательно, в документе можно реализовать управление структурой документа в соответствии с заданными правилами. Наиболее важно то, что с появлением языка JavaScript статический характер страниц HTML стал уделом прошлого. Ниже перечислены основные области применения языка JavaScript:

1. Динамическое создание документа HTML с помощью программы
2. Проверка достоверности полей форм HTML до передачи их на сервер
3. Локальный ввод информации для управления программой на языке JavaScript
4. Предоставление пользователю возможности выбора операций, выполняемых броузером
5. Вывод сообщений для пользователя (например, предостережений) в соответствующих окнах
6. Локальная обработка форм, локальный ввод информации пользователем и другие «домашние» задачи

Изучение языка JavaScript поможет при подготовке к освоению более сложного и важного языка Java. Оба языка — и JavaScript, и Java — являются полноценными языками программирования. Но при этом можно утверждать, что язык JavaScript представляет собой своего рода модель языка Java. В программе на языке Java также используются классы и их методы, но он является более сложным, чем язык JavaScript, поскольку в языке Java нужно обязательно объявлять классы и реализовывать их методы.

Важно четко разграничить области применимости каждой из этих сред программирования. Язык JavaScript никогда не рассматривался в качестве замены языка Java; в идеальном варианте его следует использовать в качестве дополнения к языку Java, позволяющего слить воедино все операционное

окружение Web-приложения и предоставить его пользователю. Java используется главным образом для решения «ответственных задач», например для разработки графических интерфейсов пользователя, тогда как язык JavaScript предназначен для связывания воедино всех строительных блоков приложения. Словом, язык JavaScript – по существу средство построения фундамента.

Требования языка JavaScript также значительно менее строги в вопросах синтаксиса и проверки типов. Компилируемая система Java базируется на понятии классов, которые реализуются путем их объявления. JavaScript реализован как интерпретатор (т.е. объектный код не генерируется) с небольшим количеством примитивных базовых типов, к которым относятся строки, а также числовой и булев типы. На основе примитивных типов можно также строить объекты, определяя их свойства с помощью оператора присваивания. Главное, что всегда нужно помнить при разработке сценариев, – это то, что Web-страницы быстро уходят в прошлое; основной упор сейчас делается на завершенные «приложения», которые объединяются в единое целое с помощью таких технологий, как Java, JavaScript, JSP, Perl и HTML.

Как поместить программу на языке JavaScript в документ HTML

Программы на языке JavaScript встраиваются в страницу текста на языке HTML. Броузер распознает программу на языке JavaScript, помещенную между начальным и конечным тегами <script>, и приступает к ее выполнению. Тег <script> HTML является контейнером, и поэтому для того, чтобы указать конец тела сценария Java, всегда требуется </script>. Сам код сценария располагается, соответственно, внутри контейнера.

Сценарии можно помещать в любом месте документа HTML, но важно помнить, что теги HTML и операторы JavaScript нельзя располагать вперемешку – иначе результат может оказаться некорректным. Однако с помощью операторов JavaScript можно динамически генерировать код HTML.

Варианты заданий к лабораторным работам по использованию JavaScript и Java апплета.

1. Реализовать калькулятор для 4 основных арифметических действий, с двумя регистрами памяти и соответствующими окнами для отображения их содержимого.
2. Реализовать 20-ти разрядный калькулятор для 4 основных арифметических действий.
3. Реализовать калькулятор работающий в 16-ричной системе исчисления.
4. Реализовать калькулятор работающий в 8-ричной системе исчисления.
5. Реализовать калькулятор с дополнительными функциями (квадратный корень, возведение в любую степень, кнопкой отображения текущего времени) и одним регистром памяти.

6. Реализовать калькулятор, умеющий работать с часами, минутами и секундами.
7. Реализовать калькулятор, умеющий рассчитывать выражения, содержащие 4 основных арифметических действия, скобки и символ подстановки значения регистра памяти.
8. Реализовать калькулятор, помнящий 10 последних результатов расчетов с возможностью возврата к любому из них.
9. Реализовать калькулятор для 4 основных арифметических действий, умеющий рассчитывать как реальные так и комплексные числа и возможностью перевода между двумя представлениями комплексных чисел.
10. Реализовать калькулятор, решающий уравнение с одним неизвестным, содержащее 3 арифметических действия (+,-,*) и скобки. Например: 3*X-3*(9+5*X)=7
11. Реализовать венгерский калькулятор (МК-61) для 4 арифметических действий, имеющий 5 стековых регистров.
12. Реализовать калькулятор, умеющий рассчитывать производную выражения, содержащего 4 арифметических действия, возведение в степень и скобки. Например: [4*X^3+5*X-(4*X^2+3)^2]` = 12*X^2+5-2*(4*X^2+3)*(8*X).
13. Реализовать калькулятор, помимо основных операций, рассчитывающий тригонометрические функции в градусах и радианах.
14. Реализовать калькулятор, для перевода: градусов в радианы; часов и долей часа в часы, минуты и секунды; декартовых координат в полярные с началом по оси X. Организовать перевод в обе стороны.
15. Реализовать статистический калькулятор, рассчитывающий математическое ожидание, дисперсию, сумму и сумму квадратов выборки до 30 цифр.
16. Реализовать калькулятор, помимо основных действий, рассчитывающий факториал, перестановки, сочетание и размещение.

Во всех калькуляторах предусмотреть контроль корректности данных на всех этапах работы.

Контрольные вопросы:

1. Что такое JavaScript?
2. Функции и основные области применения JavaScript?
3. Как встроить JavaScript в HTML страницу?
4. Как программа на JavaScript взаимодействует со страницей-носителем?
5. Основные типы данных в JavaScript?

ЛАБОРАТОРНАЯ РАБОТА № 5
НАСТРОЙКА WEB-СЕРВЕРА И РАЗРАБОТКА ДИНАМИЧЕСКИХ WEB-СТРАНИЦ НА ОСНОВЕ ТЕХНОЛОГИИ CGI

Apache – наиболее популярный WEB-сервер перенесенный на большинство операционных систем. Среди основных особенностей можно выделить:

1. мощный, удобный, надежный Web-сервер:
2. поддержка последних протоколов, в т.ч. HTTP/1.1 (RFC2616);
3. легко настраиваемый и расширяемый с использованием модулей сторонних поставщиков;
4. может быть расширен за счет написания модулей с использованием Apache module API;
5. постоянно развивающийся;
6. При работе с сервером Apache можно выделить следующие основные директории:
7. conf – содержит файлы конфигурации, из которых самым важным является httpd.conf;
8. htdocs – содержит HTML сценарии, предоставляемые клиентам узла. Этот каталог и его подкаталоги образуют Web-пространство, доступного для каждого, кто работает в Web;
9. logs – содержит данные регистрации доступа и ошибок.

Назначение Perl – помочь программисту в выполнении рутинных задач, которые для shell слишком трудны или плохо переносимы, а также чересчур заумны, сложны для кодирования на C или ином используемом языке. Мощные конструкции этого языка позволяют создавать (с минимальной затратой сил) некоторые очень эффективные специализированные решения и универсальные инструменты. Задуманный первоначально как язык для операционной системы UNIX, Perl сейчас работает практически везде, включая MS-DOS, VMS, OS/2, Macintosh и все известные разновидности Windows.

Рассмотрим пример типичной Perl программы взаимодействующей с броузером через CGI интерфейс.

```
#!c:/program files/Perl520/bin/perl -w
use strict;
use CGI qw(param);
my ($age);
$age = param("age");
print "Content-Type: text/html\n\n";
print "<html><body>";
print "Вам $age лет";
print "</body></html>";
```

Первая строка говорит о том, что программа написана на языке Perl. Для Windows систем это будет конструкция типа: #!c:/program files/Perl/bin/perl. Ключ –w указывает на то, что Perl будет выдавать предупредительные сообщения о потенциально опасных конструкциях. Рекомендуется всегда использовать этот параметр. Конструкция use strict говорит о том, что все переменные должны быть описаны в блоке my(). Оператором use мы так же можем подключать внешние модули так, как это сделано с модулем CGI в данном примере. Получение входных параметров из CGI интерфейса осуществляется функцией param модуля CGI. Выдачу результатов работы программы следует производить в стандартный выходной поток. При этом для выдачи текстовой информации используется определитель Content-Type: text/html с обязательной пустой строкой перед данными.

Клиент может запросить у веб-сервера как документ-файл с диска, так и документ, динамически формируемый некоторой внешней программой (как правило – в зависимости от данных, предоставленных пользователем при заполнении формы). Интерфейс CGI представляет собой спецификацию взаимодействия веб-сервера и внешней программы, которую веб-сервер запускает для обработки запроса. (Внешняя программа, вне зависимости от своей природы, часто называется CGI-скриптом.)

CGI определяет каким образом данные, предоставленные клиентом в запросе, передаются программе, как программа возвращает сгенерированный HTML-контент серверу, и какие переменные окружения устанавливаются сервером при запуске программы. Переменные окружения несут дополнительную информацию о сервере и запросе (например, тип сервера, IP-адрес клиента и др.).

Данные из заполненной клиентом HTML-формы могут передаваться на сервер двумя методами: GET и POST, это определяется параметром method соответствующего тэга <form method=... action=...>. В первом случае (GET) данные присоединяются после вопросительного знака в конец URL, указанной в параметре action, во втором случае - передаются в теле запроса - в секции, предназначенной для данных (следует после всех заголовокв и пустой строки). В обоих случаях данные кодируются одинаково - см. след. пункт.

При вызове CGI-программы все, что поступило в теле запроса, подается программе на стандартный ввод, а все, что находится в URL после вопросительного знака, помещается в переменную окружения QUERY_STRING. Веб-сервером данные запроса никак не интерпретируются и не преобразуются, эти задачи возложены на CGI-программу.

CGI-программа выдает содержимое ответа (как правило, HTML-контент) на свой стандартный вывод, который перехватывается веб-сервером с тем, чтобы отослать эти данные клиенту. Предварительно CGI-программа должна напечатать заголовок «Content-Type» и отделить его от данных пустой строкой. Например, вывод CGI-программы, генерирующей HTML, может выглядеть следующим образом:

Content-Type: text/html

```
<HTML>
  <BODY>
    <H1>Hello, world</H1>
  </BODY>
<HTML>
```

Конфигурирование сервера Apache

Исполнение CGI-скриптов. Для того, чтобы Apache воспринимал все файлы, находящиеся в некотором каталоге как CGI-скрипты, нужно использовать директиву

ScriptAlias /виртуальный/путь/ /путь/к/каталогу/

ScriptAlias /cgi-bin/ /usr/local/www/cgi-bin/

Рассмотрим пример настройки сервера Apache. Программное обеспечение сервера будет состоять из следующих компонент:

1. Веб сервер - Apache, версия: 2.0.43
2. Php интерпретатор - PHP, версия: 4.2.3
3. Обработчик бинарных PHP скриптов - Zend Optimizer, версия: 2.0.3
4. Perl интерпретатор - Active Perl, версия: 5.8.0
5. Mysql сервер - MySQL, версия: 3.23.53
6. Почтовый сервер - Courier Mail Server, версия: 1.54
7. Ftp сервер - Pablo’s FTP Server, версия: 1.52
8. Sendmail - Indigo Mail, версия: 2.00
9. CronTab - Cron для Windows

После того, как все необходимое программное обеспечение будет скачано необходимо создать каталог install_server и поместить туда все ПО для установки. Запустить установочный файл сервера apache_v2_0_43.msi, после чего в появившемся окошке нажать кнопку next. Далее необходимо ознакомится с лицензией, подтвердить согласие, выбрав пункт I accept the terms in the agreement, после чего снова нажать кнопку next.

В следующем диалоговом окне необходимо ввести информацию о сервере, заполнив соответствующие поля следующими данными:

Network Domain: **localhost**

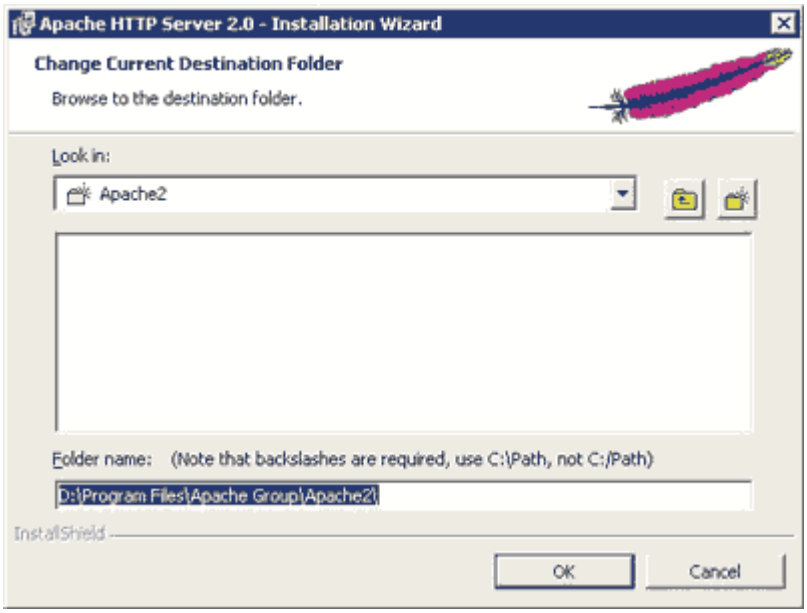
Server name: **localhost**

Administrator`s Email Address: **localhost@localhost**

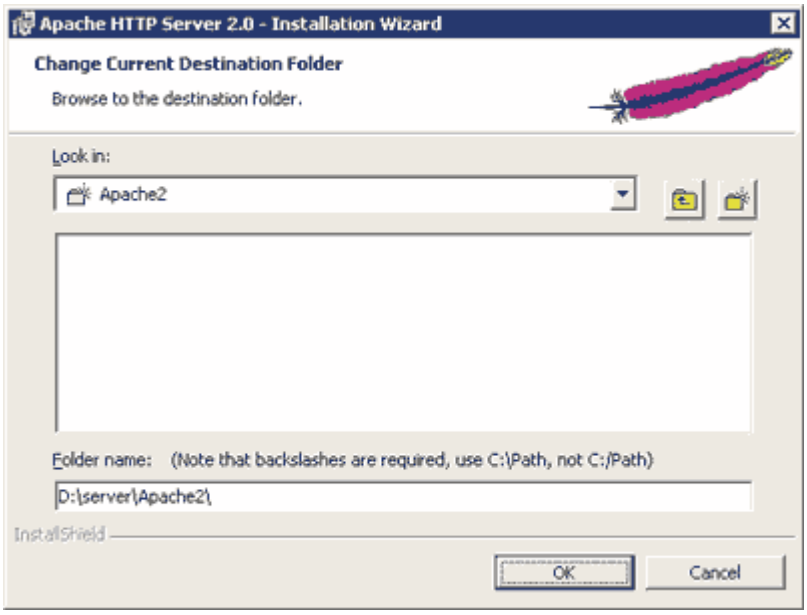
Так же выберем "**For All Users, on 80 port, as a Service -- Recommended**"

После заполнения снова необходимо нажать кнопочку next. В появившемся диалоговом окне выберем тип установки. Рассмотрим вариант Custom.

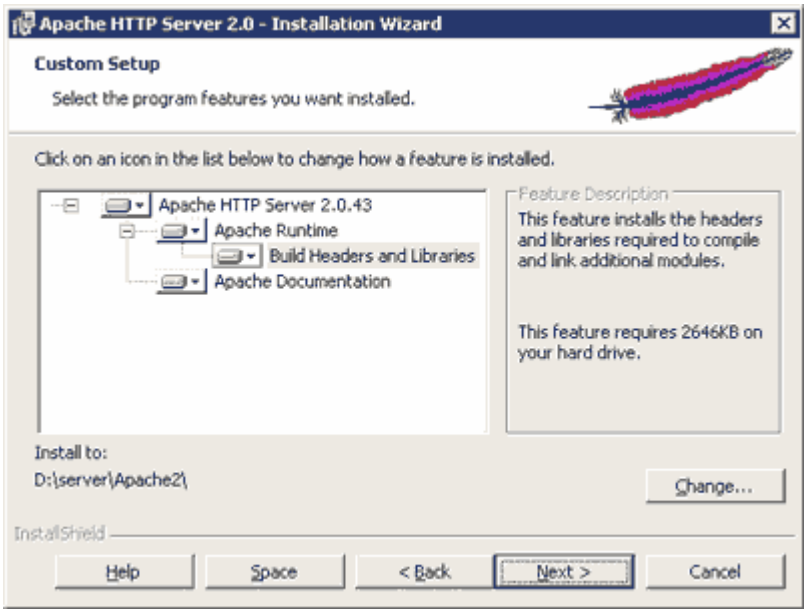
Необходимо выбрать, что будем устанавливать и в какие каталоги. Щёлкните на крестик, напротив которого написано "Build Headers and Libraries", появится меню, в ней нужно выбрать "This feature will be installed on local hard drive". Осталось выбрать папку, куда и установиться сам сервер. Щёлкаем на кнопочку "Change…", появится окно:



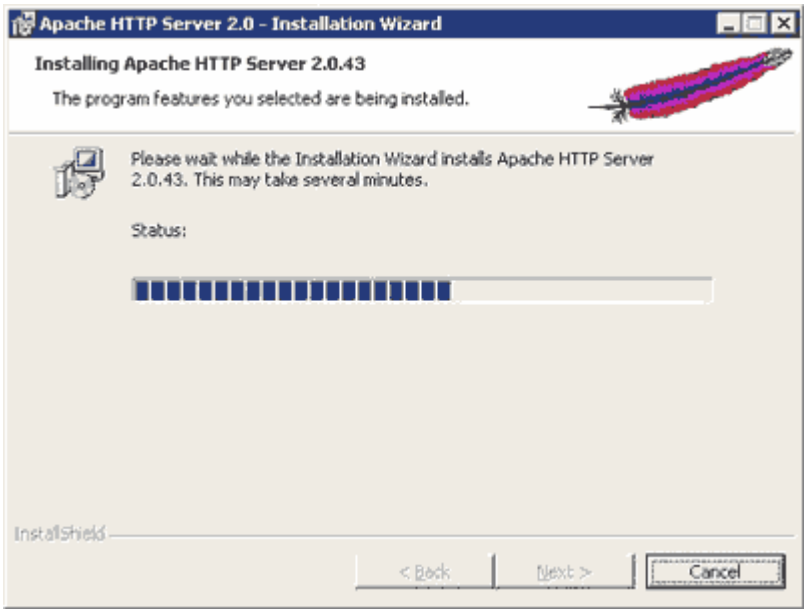
Меняем строчку "**D:\Program Files\Apache Group\Apache2**" на "**D:\server\Apache2**".



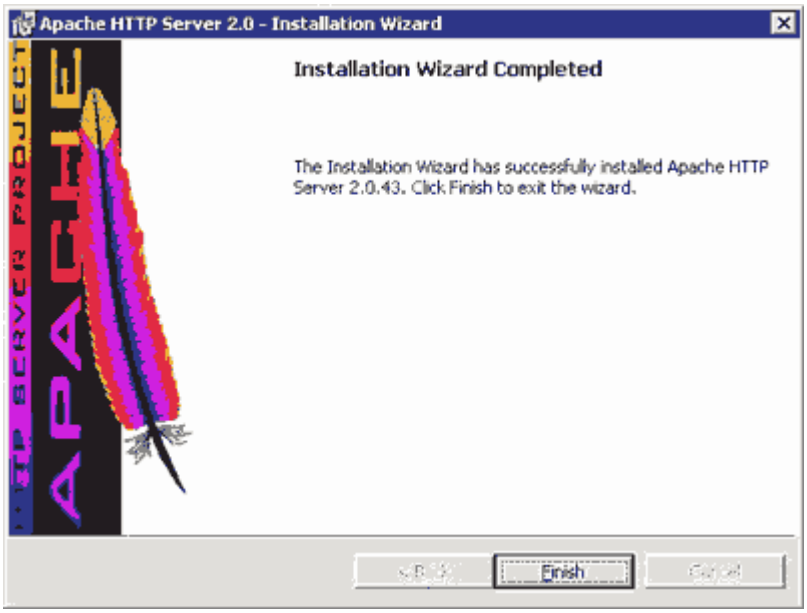
Щёлкаем на "**OK**". После щелчка по этой кнопочке появится следующее диалоговое окно:



Начинаем установку

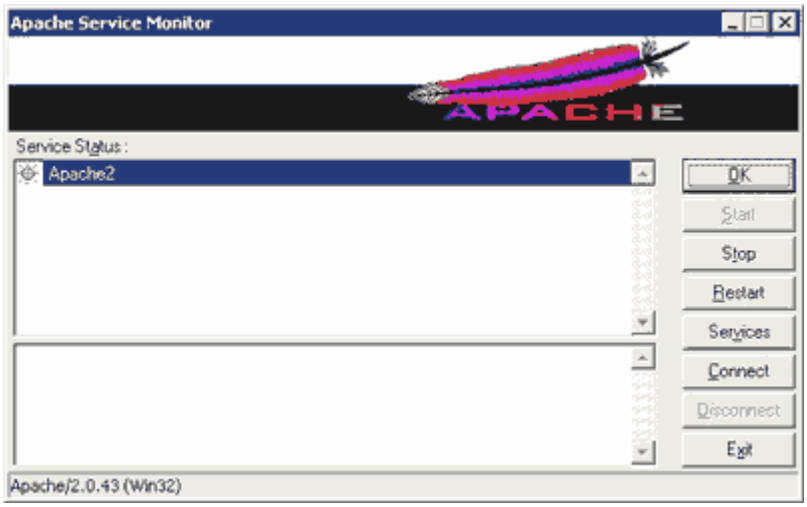


После завершения появляется следующее диалоговое окно:



Щёлкаем на кнопочку "Finish". Окошко закроется, сервер установлен. После всего этого, в правом нижнем углу у Вас появится, пёрышко с кружком, в котором виден перевёрнутый треугольник.

На картинке, перо расположено слева (оно будет появляться при каждом запуске Windows). Щёлкните на него пару раз левой клавишей своей мышки, появится окошко:



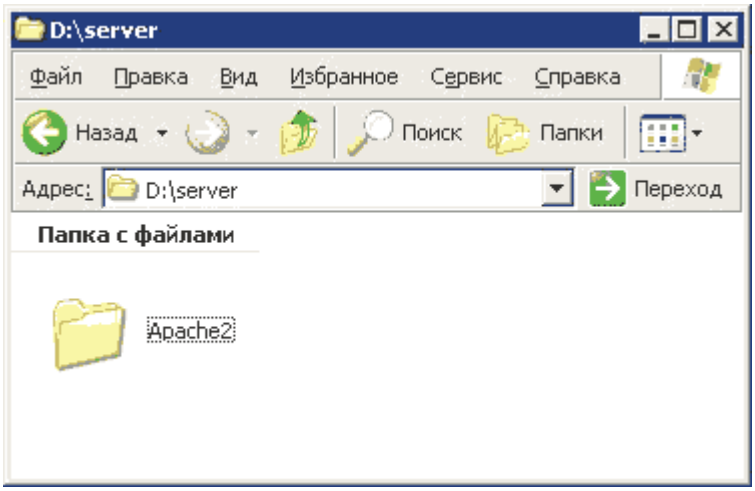
Это – небольшая панелька для управления нашим веб сервером.

Разберём эту панельку по порядку, щёлкнув по кнопочке "OK", это окошко просто закроется и ничего не произойдёт.

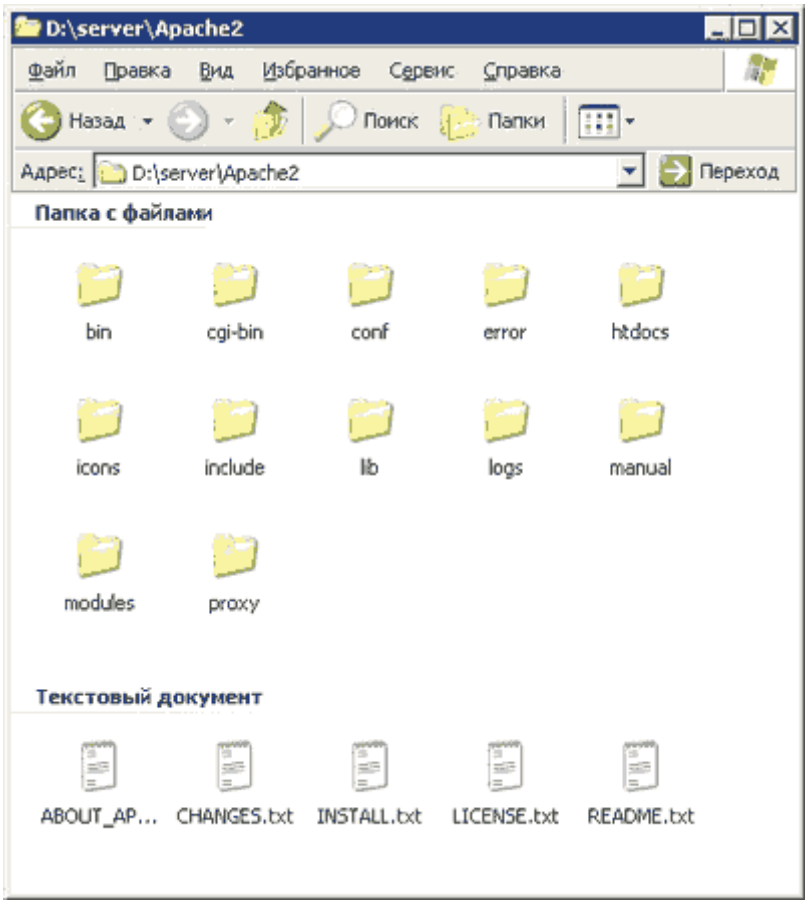
Далее следует не нажимаемая кнопка "Start", не нажимаемая она потому, что веб сервер уже запущен, а сама кнопочка означает – запустить (включить) сервер.

Следом идёт кнопка "Stop". Нажав на неё, веб сервер, отключиться.

Далее следует кнопка "Restart" необходимая для перезагрузки сервера. Нажимать ее рекомендуется после внесения изменений в конфигурационные файлы сервера.



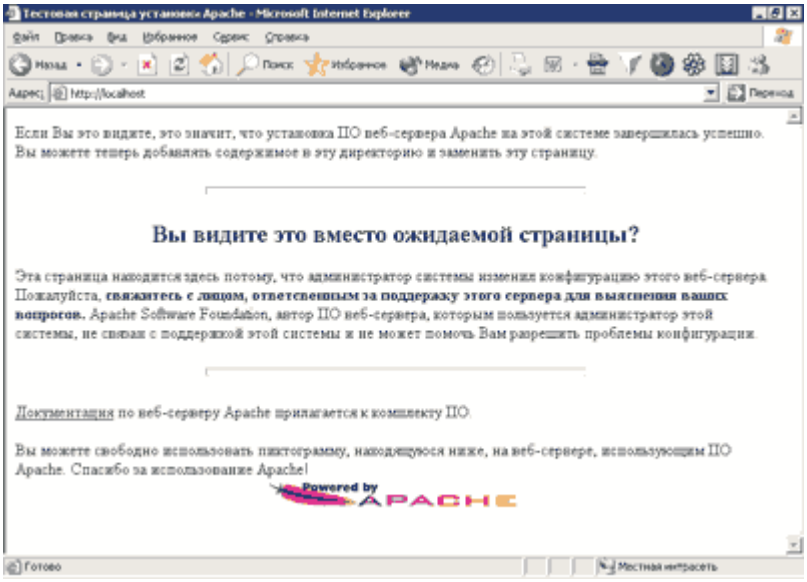
Содержимое каталога Apache2, должно выглядеть следующим образом:



Тестирование сервера

Для проверки работоспособности сервера необходимо запустить браузер, в строке адреса, ввести http://localhost (или http://127.0.0.1 – это синоним от http://localhost).

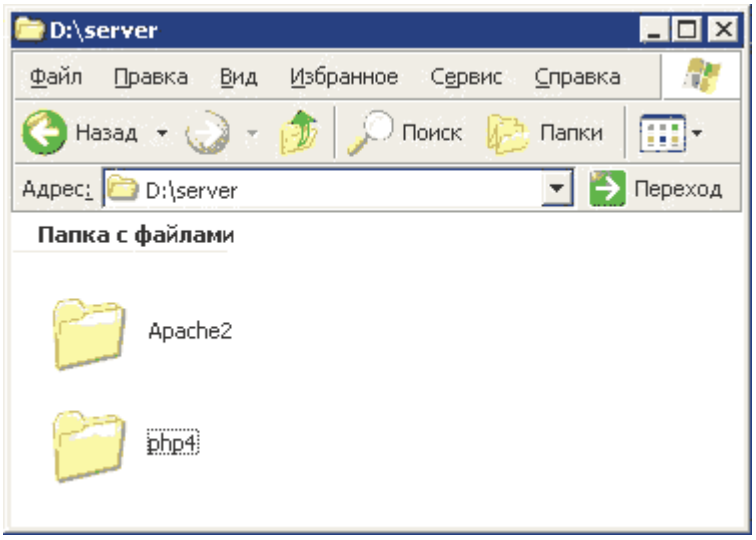
В случае, если веб сервер установлен удачно, то в браузере отображается следующий результат:



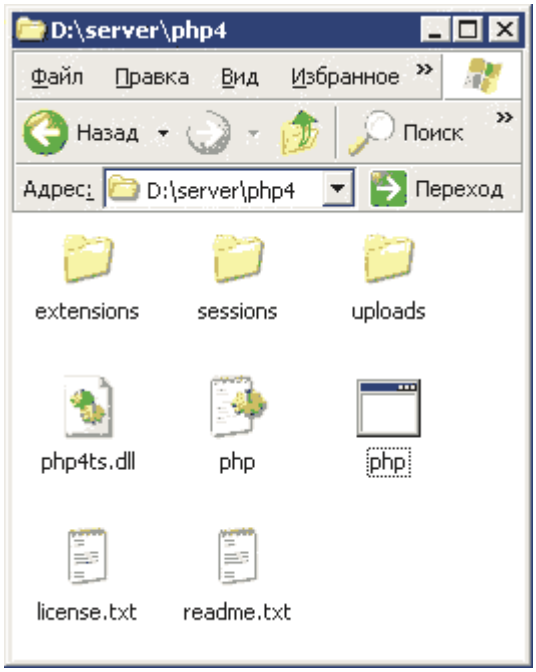
Установка надстроек для сервера

Установка PHP интерпретатора, для обработки php скриптов.

Для установки PHP интерпретатора необходимо войти в каталог с именем install_server, найти в ней архив php_vX_X_X.zip, разархивировать его в каталог php_vX_X_X, переименовать его в php4, и скопировать в папку server, т.е. папка php4, должна находиться в директории D:\server

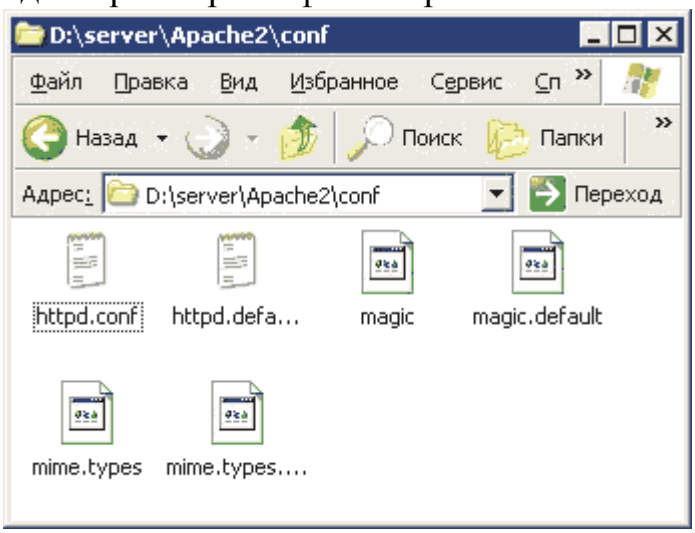


Внутренности папки php4, должны выглядеть так:



Редактирование конфигурационных файлов веб сервера.

Для редактирования конфигурационных файлов веб сервера Apache необходимо в каталоге D:\server\Apache2\conf при помощи блокнота или другого текстового редактора открыть файл httpd.conf.



В который необходимо вписать следующие строки

```
# Автор конфигурационного файла для веб сервера Apache ФИО ст-т гр ...
# Установка SSI, для файлов с расширением shtm и shtml
AddType text/html .shtm .shtml
AddOutputFilter INCLUDES .shtm .shtml

# Указываем веб серверу, что у нас есть PHP интерпретатор
ScriptAlias /php4/ "D:/server/php4/"
Action application/x-httpd-php4 "/php4/php.exe"
```

```
# Устанавливаем расширения для PHP скриптов
AddType application/x-httpd-php4 .htm .html .php .php3 .php4 .phtm .phtml
# Папка куда установлен наш веб сервер
ServerRoot "D:/server/Apache2"
PidFile logs/httpd.pid
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
HostnameLookups Off

# Порт, который слушает наш веб сервер
Listen 80

# Модули
LoadModule access_module modules/mod_access.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_module modules/mod_auth.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule cgi_module modules/mod_cgi.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
LoadModule imap_module modules/mod_imap.so
LoadModule include_module modules/mod_include.so
LoadModule isapi_module modules/mod_isapi.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule userdir_module modules/mod_userdir.so

# E-mail адрес администратора веб сервера (свой адрес)
ServerAdmin localhost@127.0.0.1

# Наш сервер:порт
ServerName localhost:80

UseCanonicalName Off

# Папка, где будут храниться наши html, php и другие файлы
DocumentRoot "D:/server/www"
```

```
# Индексные файлы, через пробел
DirectoryIndex index.htm index.html index.shtm index.shtml index.php index.php3 index.php4

# Имя встроенного конфигурационного файла
AccessFileName .htaccess

# Закрываем доступ к файлам конфигурации (.htaccess) и паролей (.htpasswd)
<Files ~ "^\.ht">
Order allow,deny
Deny from all
</Files>

# База с типами файлов
TypesConfig conf/mime.types

# Тип всех документов – текстовый
DefaultType text/plain

<IfModule mod_mime_magic.c>
MIMEMagicFile conf/magic
</IfModule>

# Файл отчёта с ошибками
ErrorLog logs/error.log

# Что записывать в файл отчёта, может принимать следующие значения:
# debug, info, notice, warn, error, crit, alert, emerg.
LogLevel warn

# Шаблон записи строки в файл отчёта
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Файл отчёта со списком всех доступов к веб серверу
CustomLog logs/access.log common

ServerTokens Full
```

```
# Подпись веб сервера (On - включена, Off - отключена, EMail - показывать e-mail администратора сервера)
ServerSignature On

# Виртуальные папки
# Например:
# Alias /icons/ "d:/server/Apache2/icons/"
# Зайдя на http://localhost/icons/ мы увидем содержимое папки d:/server/Apache2/icons/
Alias /icons/ "D:/server/Apache2/icons/"

# Папка длс CGI, для PERL скриптов
ScriptAlias /cgi-bin/ "D:/server/cgi-bin/"
<Directory "D:/server/cgi-bin">
AllowOverride None
Options None
Order allow,deny
Allow from all
</Directory>

AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
# Выбираем иконки для различных форматов
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
```

```
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

# Иконка для неизвестных форматов
DefaultIcon /icons/unknown.gif

# Если выводится содержимое какой-либо папки, считываются файлы описания:
ReadmeName README.html
HeaderName HEADER.html

# Файлы для игнорирования, при выводе листинга директории
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

AddEncoding x-compress Z
AddEncoding x-gzip gz tgz

# Наша кодировка
AddDefaultCharset WINDOWS-1251

# CGI, PERL скрипты
AddHandler cgi-script .cgi .pl
```

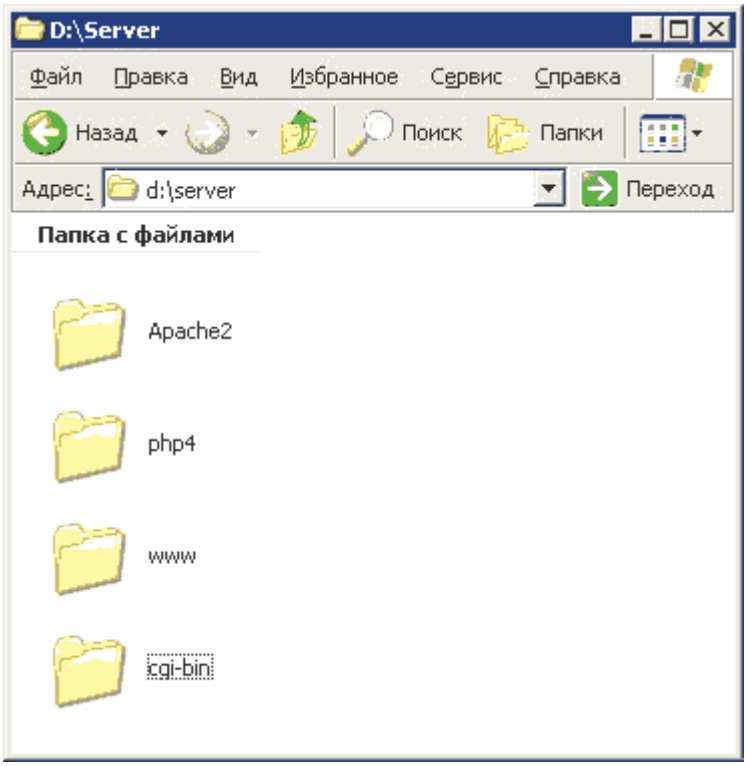
После внесения изменения файл необходимо сохранить.

После выполнения этой процедуры конфигурационный файл сервера отредактирован и настроен.

Далее следует создать каталоги www и cgi-bin в директории D:\server

В каталоге www, будут храниться html, php и другие файлы. В каталоге cgi-bin, будут храниться perl скрипты.

После этого каталог веб сервера, должен выглядеть следующим образом:



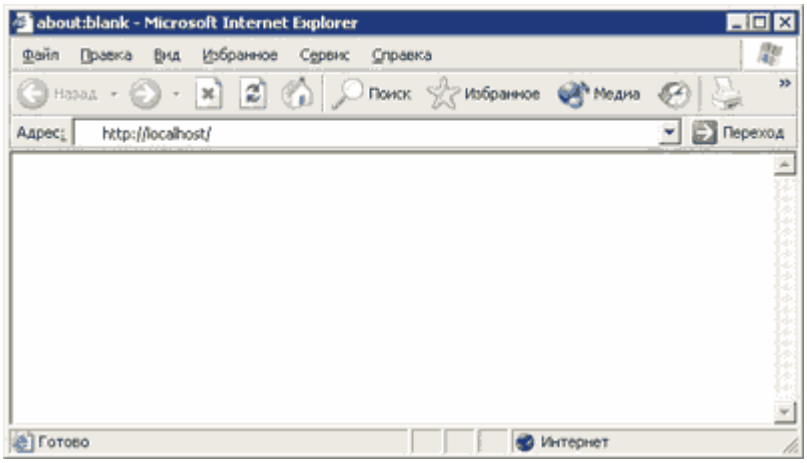
После внесений изменений веб сервер необходимо перезагрузить.

Тестирование веб сервера

Создадим в каталоге D:\server\www текстовый документ с именем index.php и впишем в него следующий код:

```
<?
echo "http://adi-eki.org.ua ";
?>
```

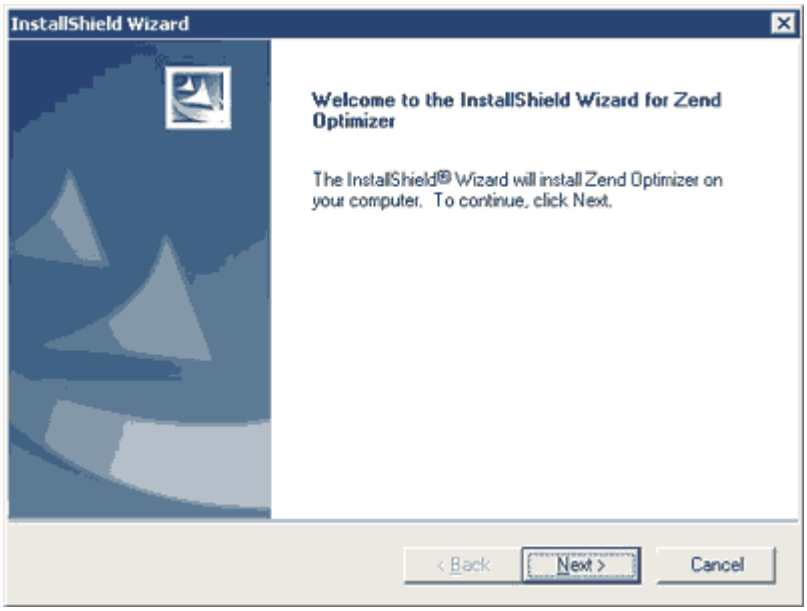
Сохраним файл, откроем браузер и в адресной строке введем адрес http://localhost/



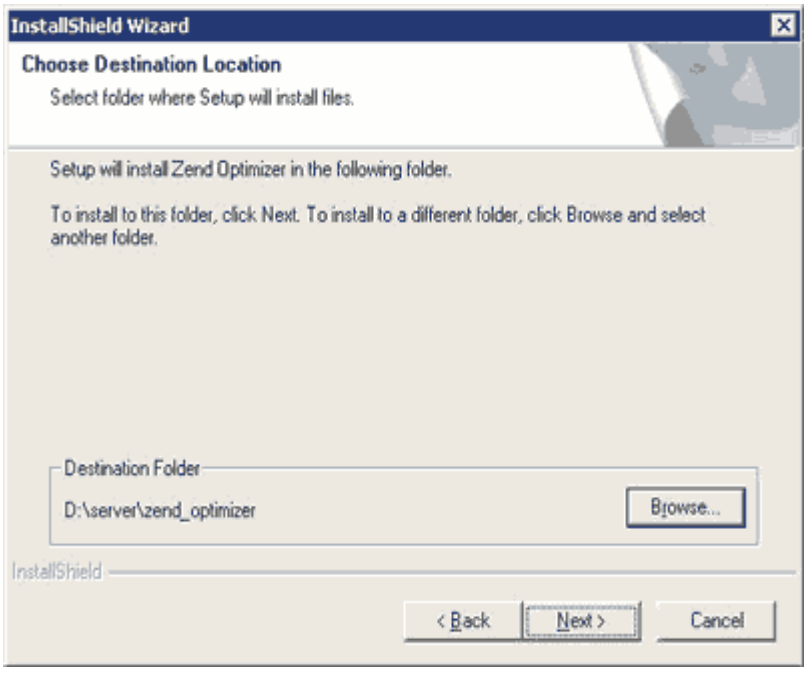
Если PHP интерпретатор установлен удачно, то в окне браузера появится следующая строка http://www.adi-eki.org.ua

Установка поддержки бинарных php скриптов.

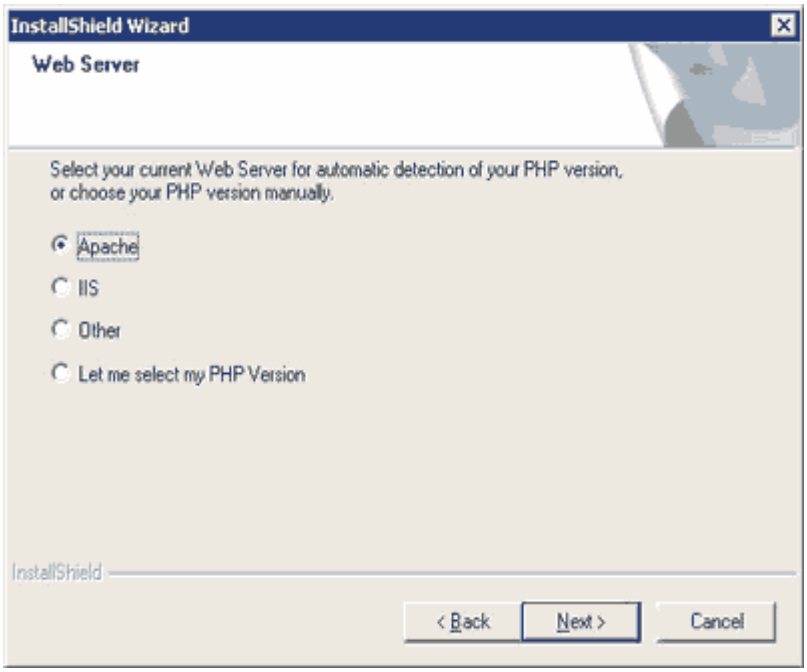
Открываем нам уже каталог install_server, находим архив zend_optimizer_v2_0_3.zip, разархивируем его в каталог zend_optimizer_v2_0_3, открываем каталог, запускаем файл zend_optimizer_v2_0_3.exe, идёт подготовка к установке, появляется вот такое окошко:



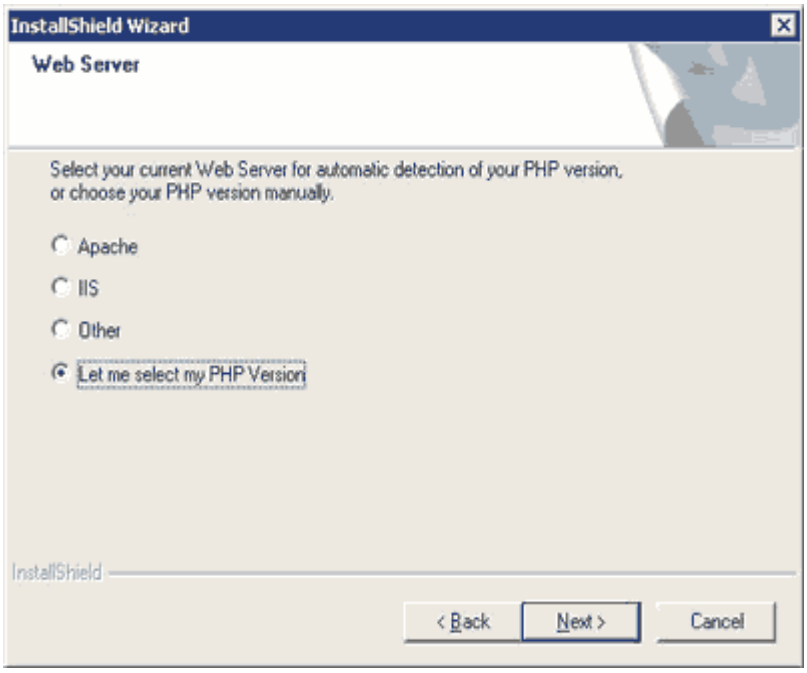
Щёлкаем на "Next". В следующем диалоговом окне указываем путь для установки D:\server\zend_optimizer:



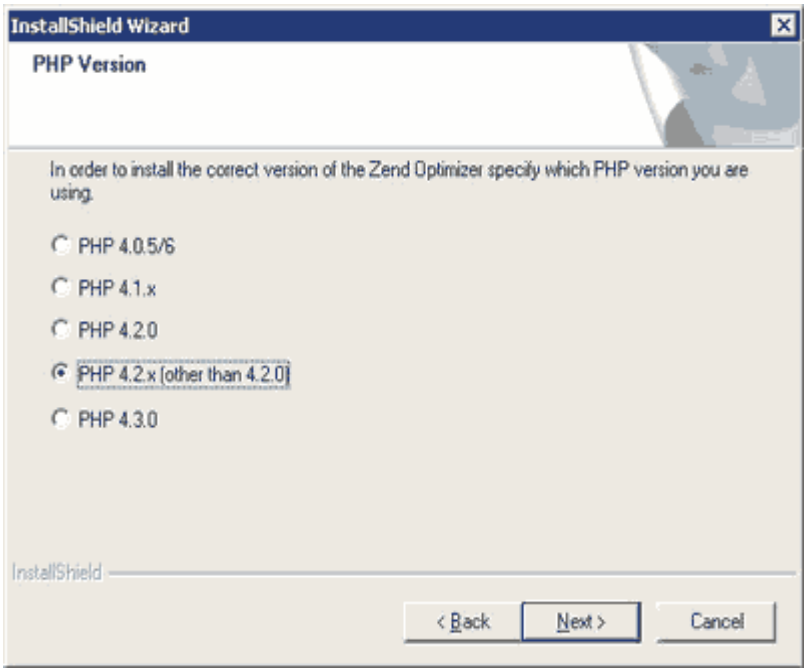
После нажатия на "Next" появляется следующее диалоговое окно:



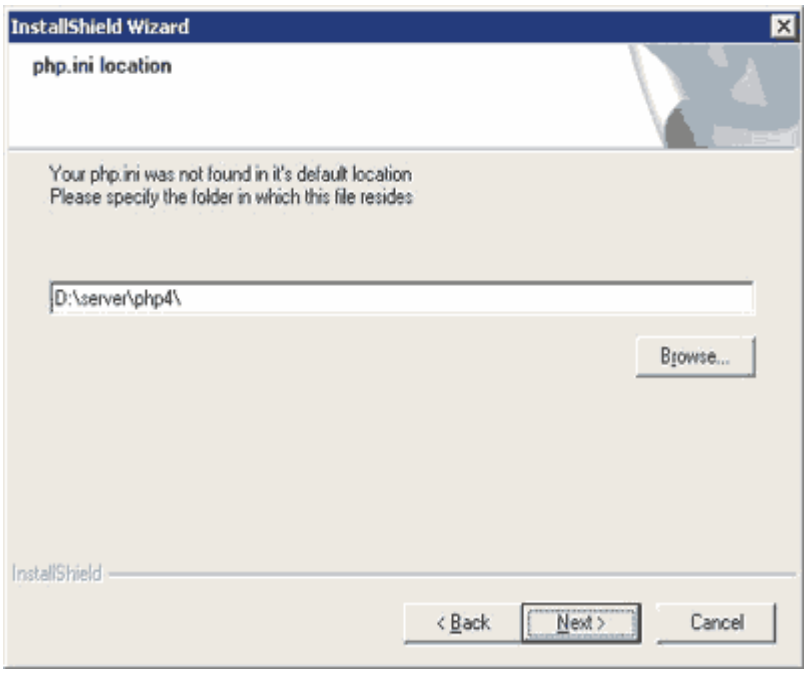
Необходимо выбрать "Let me select my PHP Version":



В следующем диалоговом окне необходимо выбрать пункт PHP 4.2.x (other than 4.2.0)



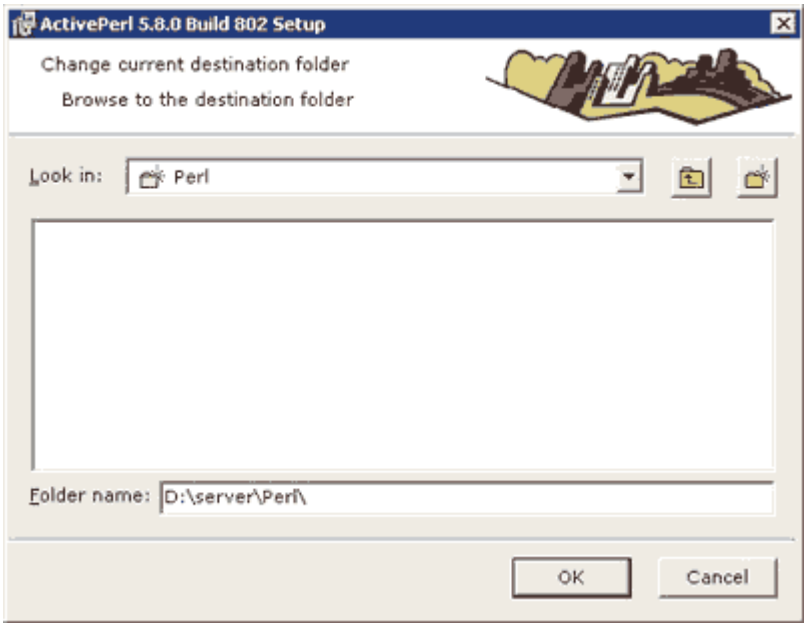
Вписываем путь расположения ini файлов сервера php "D:\server\php4\"



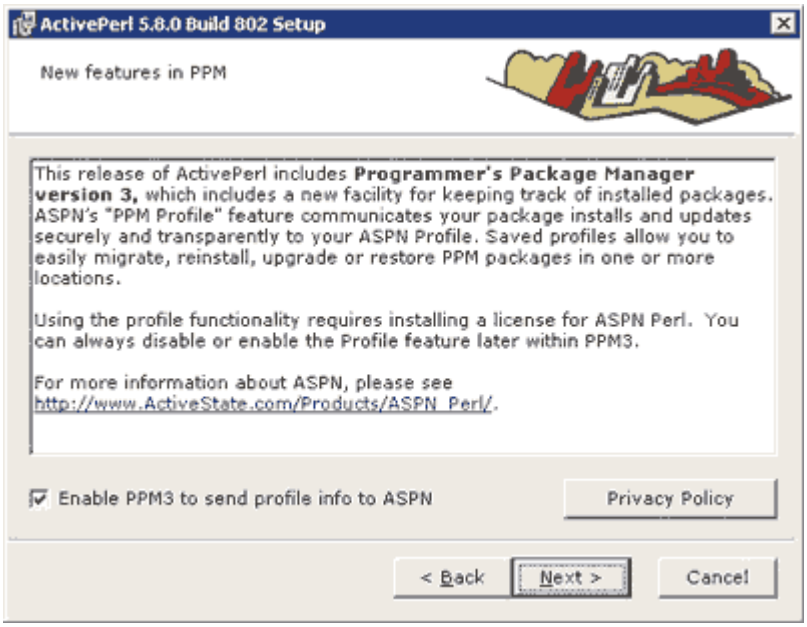
Приступаем к установке ...

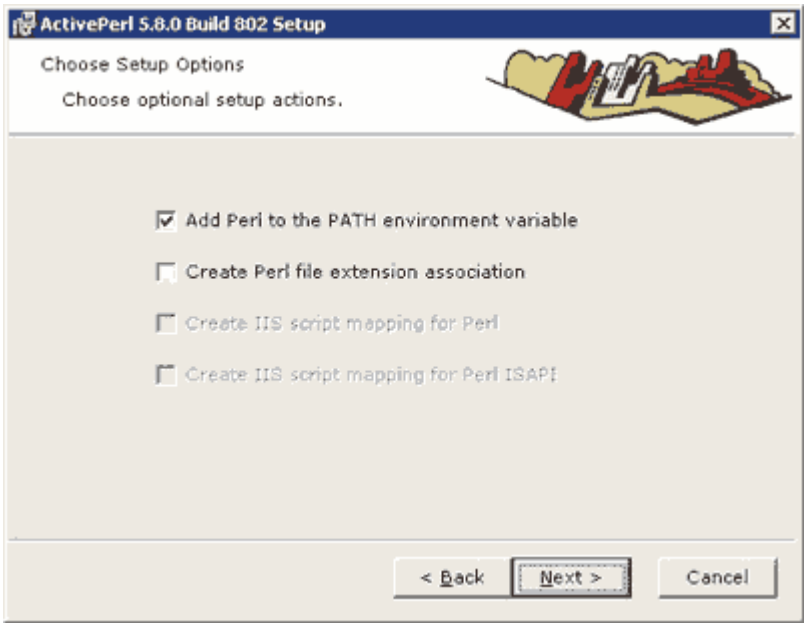
Установка PERL интерпретатора, для обработки perl скриптов.

Из каталога install_server разархивируем архив active_perl_vX_X_X.zip в каталог active_perl_vX_X_X, запускаем файл active_perl_vX_X_X0.msi и устанавливаем интерпретатор в каталог "D:\server\Perl\"

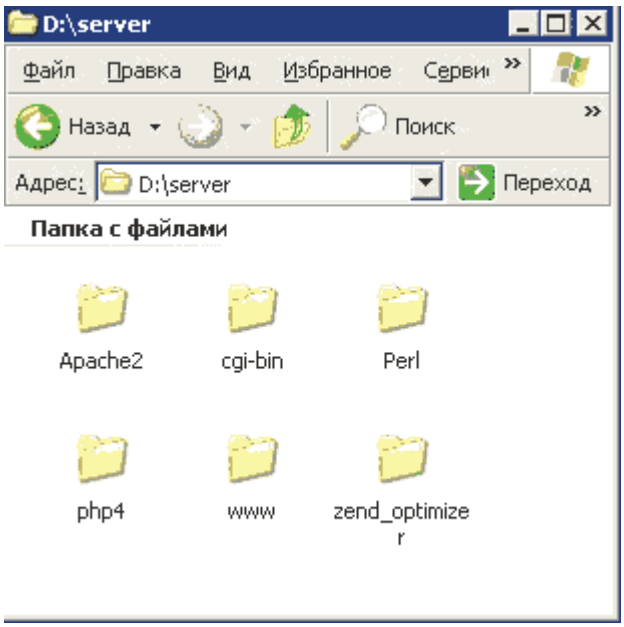


В процессе установки необходимо отметить пункт "Enable PPM3 to send profile info to ASPN", снять выбор пункта Create Perl file extension association





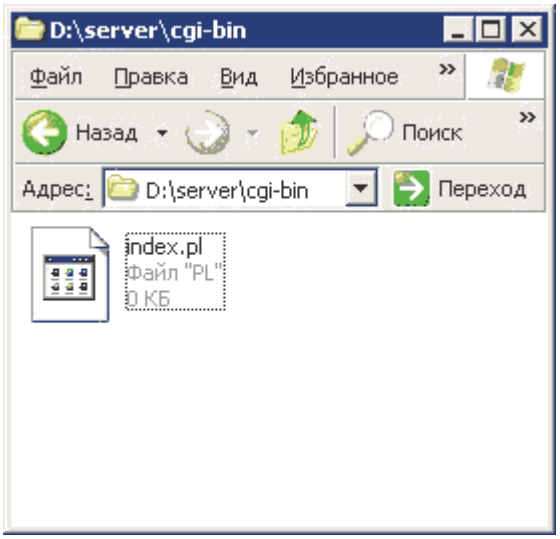
После установки Perl интерпретатора каталог сервера, а именно D:\server выглядит следующим образом:



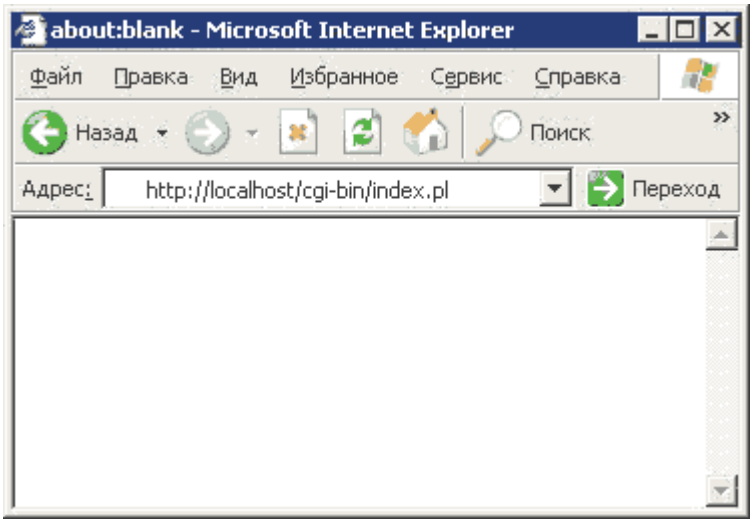
Тестирование PERL интерпретатора.

Для тестирования интерпретатора необходимо открыть каталог D:\server\cgi-bin, создать в каталоге текстовый файл с именем index.pl, открыть блокнот или любой другой текстовый редактор и вписать следующий код:

```
#!perl
print "Content-type: text/html\r\n\r\n";
print "http://adi-eki.org.ua";
```



Сохранить файл и в браузере ввести строку http://localhost/cgi-bin/index.pl



После установки и настройки сервер готов к работе. Следует учесть, что:

1. веб сервер, php, perl интерпретаторы, обработчик бинарных php скриптов, запускаются автоматически, вместе с загрузкой самой операционной системой Windows;
2. первая строчка в perl скриптах (это файлы с расширением pl и cgi, например: file.pl, example.cgi) должна быть:

   #!perl

   или

   #!d:/server/perl/bin/perl.exe

   или просто

   #!d:/server/perl/bin/perl
3. php код работает в файлах с расширениями: htm, html, php, php3, php4, phtm, phtml (примеры файлов: file.htm, example.html, super.php, ultra.php3, dir.php4, country.phtm, city.phtml)

Варианты заданий:

1. Написать программу, которая получает в качестве входных параметров имена файлов, номера строк, набор символов. Посчитать все вхождения указанных символов в указанных файлах и вывести результат.
2. Написать программу, которая получает в качестве входных параметров адреса хостов. Далее выполняет команду ping для указанных машин и выводит форматированный результат тестирования.
3. Написать программу, которая получает в качестве входных параметров имена файлов. Далее по каждому файлу выводит подробную информацию (размер, дата создания, атрибуты и т.д.).
4. Написать программу, которая производит операцию «Upload» (т.е. загрузку указанного файла на сервер).
5. Написать программу, которая получает в качестве входного параметра имя файла. Далее программа должна передать указанный файл от сервера к клиенту.
6. Написать программу, которая получает в качестве входного параметра имя файла и две подстроки. Далее необходимо произвести замену первой подстроки второй в указанном файле и вывести кол-во произведенных замен.
7. Написать программу, которая получает в качестве входного параметра имя файла. Далее производит суммирование всех чисел в указанном файле и выводит результат.
8. Написать программу, которая получает в качестве входных параметров шаблон и имя директории. Далее программа должна вывести все файлы удовлетворяющие указанному шаблону (допускается использование символов * и ?).
9. Написать программу, которая реализует функцию счетчика посещения страниц.
10. Написать программу, которая в качестве входного параметра получает код на любом языке программирования (C, Java, Perl, …). Далее производит раскраску (выделение цветом) основных операторов (не менее 10) и результат возвращает клиенту.
11. Написать программу, которая реализует функцию on-line переводчика. Программа получает в качестве входного параметра текст, а возвращает его перевод. Словарная база должна насчитывать не менее 50 слов.
12. Написать программу, которая реализует функцию поиска. В качестве входного параметра передаются ключевые слова. Далее программа сканирует все файлы текущей директории на предмет совпадения искомых выражений. Результат оформляется в виде ссылок на найденные файлы.

13. Написать программу, которая производит проверку веденной пользователем формы (ФИО, возраст, адрес, телефон, e-mail) на корректность. Результатом должен быть шаблон, заполненный указанными пользователем данными.
14. Написать программу, которая получает в качестве входных параметров имена текстовых файлов. Необходимо произвести объединение этих файлов в один результирующий и передать этот файл клиенту. При объединении одинаковые строки не должны повторяться.
15. Написать программу, которая реализует одну из функций работы системы WEB-почты, а именно получение списка новых сообщений.

Контрольные вопросы:

1. Назначение и принцип работы сервера Apache.
2. Назначение CGI протокола.
3. Основные отличия клиентских запросов GET и POST.
4. Что такое регулярные выражения и область их применения.
5. «Жизненный цикл» CGI приложения.
6. Основные методы написания безопасных CGI скриптов.