Федеральное государственное образовательное учреждение высшего профессионального образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ

Кафедра вычислительных машин, комплексов, систем и сетей

Н.И. РОМАНЧЕВА

назад

ИНТЕРНЕТ-ТЕХНОЛОГИИ в ГА

ПОСОБИЕ

к выполнению лабораторных работ №1,2

для студентов 5 курса
специальности 230101
дневного обучения

Москва- 2006

Рецензент доктор техн. наук, доцент А.А.Егорова

Романчева Н.И.

ИНТЕРНЕТ-ТЕХНОЛОГИИ В ГА: Пособие к выполнению лабораторных работ № 1,2. - М.: МГТУ ГА, 2006.- 32 с.

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

Рассмотрены и одобрены на заседаниях кафедры 21.06.2006г. и методического совета по специальности 230101.

С О Д Е Р Ж А Н И Е

1 ОСНОВНЫЕ ТРЕБОВАНИЯ И ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

Настоящее пособие предназначено для студентов специальности 230101, выполняющих лабораторные работы по дисциплине "Интернет-технологии в ГА". В пособие включены материалы по лабораторным работам № 1,2.

Продолжительность каждой лабораторной работы - 4 часа.

Целью проведения лабораторных работ является закрепление основных теоретических положений, изложенных в лекциях на примере широко используемых в ГА технологий FTP, telnet и PHP.

В процессе выполнения лабораторных работ студенты должны освоить приемы и методы:

- работы на FTP-серверах и TELNET;
- работы с PHP-машиной,

Лабораторная работа состоит из следующих этапов:

1) домашняя подготовка;

2) выполнение работы на компьютере в соответствии с заданием;

3) сдача выполненной работы преподавателю на персональном компьютере;

4) распечатка результатов работы на принтере;

5) оформление отчета;

5) защита лабораторной работы.

В процессе домашней подготовки студент:

- изучает лекционный материал, материалы по темам данного пособия и дополнительной литературы,
- знакомится с заданием на выполнение лабораторной работы;
- готовит отчет по выполнению лабораторной работы (пункты, отмеченные знаком *).

Выполнение лабораторной работы производится во время занятий в классе ИВЦ МГТУГА в присутствии преподавателя. В процессе выполнения лабораторной работы студент последовательно выполняет задание. По завершению работы - демонстрирует преподавателю результаты.

Сдача работы преподавателю на персональном компьютере заключается в демонстрации выполненной работы и выполнении непосредственно при преподавателе индивидуального дополнительного задания.

После приема преподавателем лабораторной работы на ПК студент:

- сохраняет результаты лабораторной работы на дискете, выданной преподавателем, в каталоге со своей фамилией;
- распечатывает результаты на принтере на подготовленных листах формата А4.

Отчет по каждой лабораторной работе должен содержать:

- название работы*;
- цель лабораторной работы*;
- задание на выполнение лабораторной работы*;
- краткие комментарии по выполнению лабораторной работы*;
- распечатки файлов результатов, подписанные преподавателем.

Защита лабораторной работы преподавателю проводится по контрольным вопросам и при наличии оформленного отчета (распечатки должны быть приклеены). После защиты лабораторной работы делается соответствующая запись на отчете студента.

В соответствии с Положением МГТУ ГА о зачетах и курсовых экзаменах студент, не защитивший 2-х работ, не допускается к выполнению следующей лабораторной работы.

2 ЛАБОРАТОРНАЯ РАБОТА №1
РАБОТА С УДАЛЕННЫМ КОМПЬЮТЕРОМ НА FTP-СЕРВЕРАХ, ИСПОЛЬЗОВАНИЕ СЕРВИСА TELNET ДЛЯ ДОСТУПА К УДАЛЕННОМУ КОМПЬЮТЕРУ

2.1. Цель лабораторной работы

Целью данной работы является получение практических навыков и умений работы на FTP-серверах, использования сервиса telnet для доступа к удаленному компьютеру, тестирования соединений в сети Интернет

2.2. Задание на выполнение лабораторной работы

1. Подключитесь к любому FTP-серверу двумя способами: через командную строку и с помощью программы CuteFTP.
2. Выберите нужный Вам каталог.
3. Выгрузите один или несколько файлов на Ваш компьютер в созданную папку VM.
4. Войдите в текстовый редактор.
5. Откройте записанный файл и просмотрите его.
6. Обратитесь по адресу telnet://fedworld.gov и загрузитесь под именем NEW.
7. Ответьте на несколько вопросов о себе. После этого программа пришлет по адресу вашей электронной почты небольшое справочное руководство.
8. Изучив руководство, выйдите на главное меню.
9. Для продолжения работы с FedWorld выберите из главного меню Option 1.

10. Выберите букву J (Job- работа) для получения сведений о рабочих местах в федеральных органах.

11. В следующем меню выберите букву Z (Zone) для выбора рабочих мест по отдельным штатам.

12. Укажите нужный вам штат.

13. Проведите поиск нужной вам работы.

14. Просмотрите описание команд, обращая внимание в первую очередь на команды действий.

15. Запустите программу Tracert.exe и проследите пути следования IP-пакетов от вашего компьютера до любого другого.

16. Определите свой IP-адрес, нажав кпопку Пуск/Выполнить и Ctrl+V.

17. Выполните тестирование соединения при помощи программы Ping.exe с каким - нибудь web-сервером по имени, а затем по его IP- адресу.

2.3. Основные теоретические сведения

Для получения файлов, хранящихся на различных компьютерах Internet, пользователи сети прежде всего используют программы, основанные на протоколе FTP.

FTP - *File Transfer Protocol* - протокол передачи файлов, определяющий правила передачи файлов в сети с одного компьютера на другой.

Для работы с FTP на удаленном компьютере необходимо ввести пароль и (или) назвать себя (свое сетевое имя). В качестве имени пользователя при работе с FTP-серверами используется слово anonymous или ftp, а в качестве пароля берется адрес электронной почты.

Доступ должен быть как минимум типа dial-up (по вызову). Для запуска FTP-клиента, нужно подать команду ftp с указанием имени рабочей машины, с которой вы хотите провести сеанс.

Синтаксис команды ftp следующий:

ftp [-name] [hostname]

Параметры для пересылаемых файлов:

-d [level] - Переход в режим отладки.

-f <filename> - Выполнять только команды файла filename.

-g - Блокировка автоматического расширения имени файла.

-h <filename> - Указать файл конфигурации.

-i - Блокировка приглашений для групповых переносов файлов.

-m - Включить программу more.

-n - Блокировка режима автоматической регистрации.

-p <filename> - Выполнить команды, содержащиеся в файле filename. Эти команды выполнятся сразу после того, как вы зарегистрируетесь.

-r - Отключить переадресацию вывода.

-s - Отключить переключение слэша.

-v - Отображать любые сообщения удаленного компьютера.

Команды для передачи файлов:

account [password] - Получить пароль доступа к дополнительным ресурсам сервера FTP.

ascii ASCII - режим передачи данных.

bell - Переслать данные и издать писк.

bget - Бинарный режим передачи данных. Аналог команды get.

binary - Бинарный режим передачи данных.

bput Переслать файл в бинарном режиме. Аналог команды put.

bye - Закончить выполнение ftp.

cd - Изменить каталог на удаленном компьютере.

close - Закрыть соединение с сервером FTP и выйти в DOS.

delete - Уничтожить файл на удаленном компьютере.

debug [mode] - Активизировать режим отладки, то есть ставить перед каждой командой, посланной на удаленный компьютер символы ->.

dir [other_directory][my_file] - Распечатать содержимое локального каталога на удаленном компьютере. Вы можете сохранить полученную таким образом информацию в файле на своем компьютере. Если Вы ввели эту команду без аргумента other_directory, то на удаленном компьютере будет выведен текущий каталог. Если отсутствует аргумент my_file, то вся информация отобразится на экране вашего монитора.

get other_file [my_file] - Получить файл с удаленного компьютера и сохранить его на локальном компьютере. Если у этой команды

отсутствует аргумент, то имя сохраняемого файла на локальном компьютере будет таким же, каким оно было на удаленном.

glob - Посредством этой команды вы можете оперировать расширениями файлов, то есть использовать команды mdelete, mget и mput не только вместе с именами файлов, но и с их расширениями. Вы можете применять стандартные символы (* и &) в расширениях передаваемых файлов.

hash - Активизировать режим печати символов # при передаче блоков, размер каждого из которых равен 1024 байта.

help [command] - Отобразить описание команды.

interactive - Активизировать режим выдачи сообщений во время приема или передачи файла.

lcd [directory] - Перейти в другой каталог локального компьютера. Если у этой команды отсутствует аргумент, то вы перейдете в каталог по умолчанию.

lls [directory] - Просмотреть каталог локального компьютера.

ls [other_directory][my_file] - Отобразить содержимое каталога удаленного компьютера. Если у команды нет аргумента other_directory, то отобразится содержимое каталога по умолчанию. Если же отсутствует аргумент my_file, то отобразится файл, в который будет помещена информация с удаленного компьютера. Если вместо последнего аргумента стоит дефис, то вся информация с удаленного компьютера будет выведена на экран вашего монитора.

mdelete [other_files] - Уничтожить файлы other_files удаленного компьютера.

mdir other_files my_file - Распечатать локальные файлы other_files на удаленном компьютере.

mget other_files - Найти на удаленном компьютере файлы other_files, расшифровать и активизировать команду get для переноса этих файлов в рабочий каталог локального компьютера.

mkdir name_directory - Создать каталог на удаленном компьютере.

mls other_files my_files - Отобразить содержимое файлов удаленного компьютера.

mode [name_mode] - Активизация режима переноса файлов в определенное место. По умолчанию установлен режим stream.

more - Включить режим more, то есть через паузу разбивать содержимое каталогов на части. Почти как конвейер в UNIX.

mput files - Найти и расшифровать локальные файлы files и запустить команду put для переноса этих файлов в рабочий каталог удаленного компьютера.

noninteractive - Не выдавать сообщения во время пересылки или приема файлов.

open host [port] - Соединиться с сервером FTP.

prompt - Показывать интерактивные сообщения.

put my_file [other_file] - Поместить локальный файл my_file на удаленный компьютер. Если у этой команды отсутствует аргумент other_file, то будет использован исходный файл.

pwd - Распечатать имя текущего каталога на удаленном компьютере.

quit - Аналог команды bye.

quote arg1 arg2 ... - Передать аргументы arg1 arg2 ... на сервер FTP и получить только код ответа.

recv other_file [my_file] - Аналог команды get.

remotehelp [name_command] - Получить список доступных команд удаленного сервера FTP.

rename old_name new_name - Дать другое имя файлу old_name удаленного компьютера.

rm other_file - Аналог команды delete.

rmdir name_directory - Стереть каталог name_directory на удаленном компьютере.

send my_file [other_file] - Аналог команды put.

sendport - Активизация режима команд port. Это позволяет ускорить пересылку файлов. Если port не работает, то по протоколу передачи файлы поступят на порт данных по умолчанию.

slashflip - Изменить режим смены слэша.

status - Отобразить состояние программы ftp в данный момент времени.

struct [name_struct] - Эта команда позволяет установить соответствие между структурой файла и указанным именем. В установке по умолчанию имя структуры есть file.

type [name_type] - С помощью этой команды вы можете установить тип ascii для текстов и тип binary или image для графики. Если аргумент у этой команды отсутствует, то Вы увидите тип по умолчанию, то есть ascii.

user name_user [password][access] - Весьма полезная команда. Вы сообщаете серверу FTP, кто вы есть. Если аргумент у этой команды отсутствует, то Вы увидите запрос на ввод пароля. Если указан только аргумент access, то после того, как Вы зарегистрируетесь, можно будет воспользоваться командой доступа account.

verbose -Активизация так называемого режима сообщений, то есть режима при котором Вы можете получать полную информацию с сервера FTP. Этот режим активизирован по умолчанию.

? [command] - Аналог команды help.

2.3.1. Программа CuteFTP

CuteFTP - FTP-клиент, который позволяет пользователям использовать возможности протокола FTP без необходимости знать все его подробности. Эта программа FTP имеет прекрасный интерфейс и множество дополнительных возможностей.

Перечислим некоторые из них:

- *Queuing of transfers* - передача по очереди. Возможно выбрать несколько файлов для перекачки, которые будут переписываться один за другим, представляя собой очередь.
- *Resume downloading* - поддержка докачки файлов после обрыва связи.

- *Resume uploading* - поддержка перекачки файлов после обрыва связи.
- *Directory comparison feature* - возможность сравнения реального каталога и каталога на удаленном ПК.
- *Macro record/playback scripting* - поддержка макро-команд.
- *Keep Alive "NOOP" command* - эта функция поддерживает соеденение с сервером около 30 минут пока CuteFTP простаивает.
- *Site Manager* - проводник для работы с закладками на всевозможные FTP - узлы.

Наиболее важной функцией этой программы является возможность докачки файлов. Вид основного окна программы CuteFTP представлен на рисунке2.1.

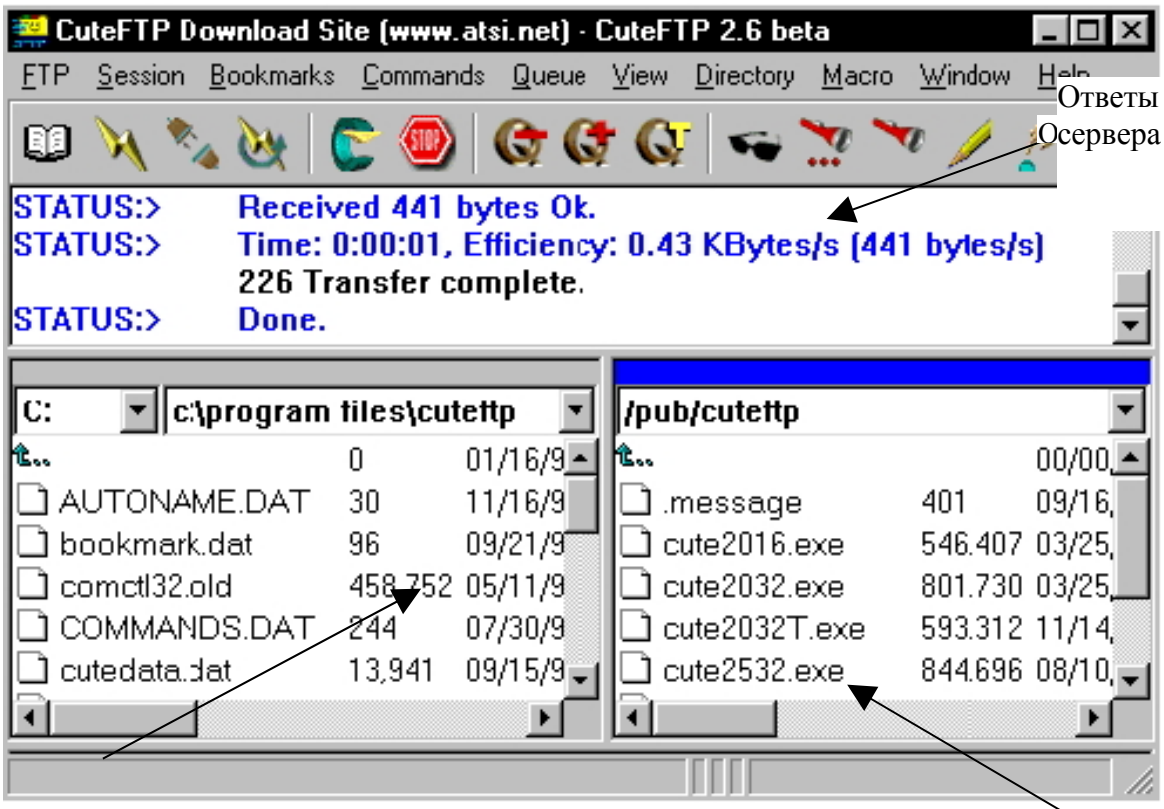

Рисунок 2.1- Основное окно программы

Вид меню программы CuteFTP представлен на рисунке 2.2.

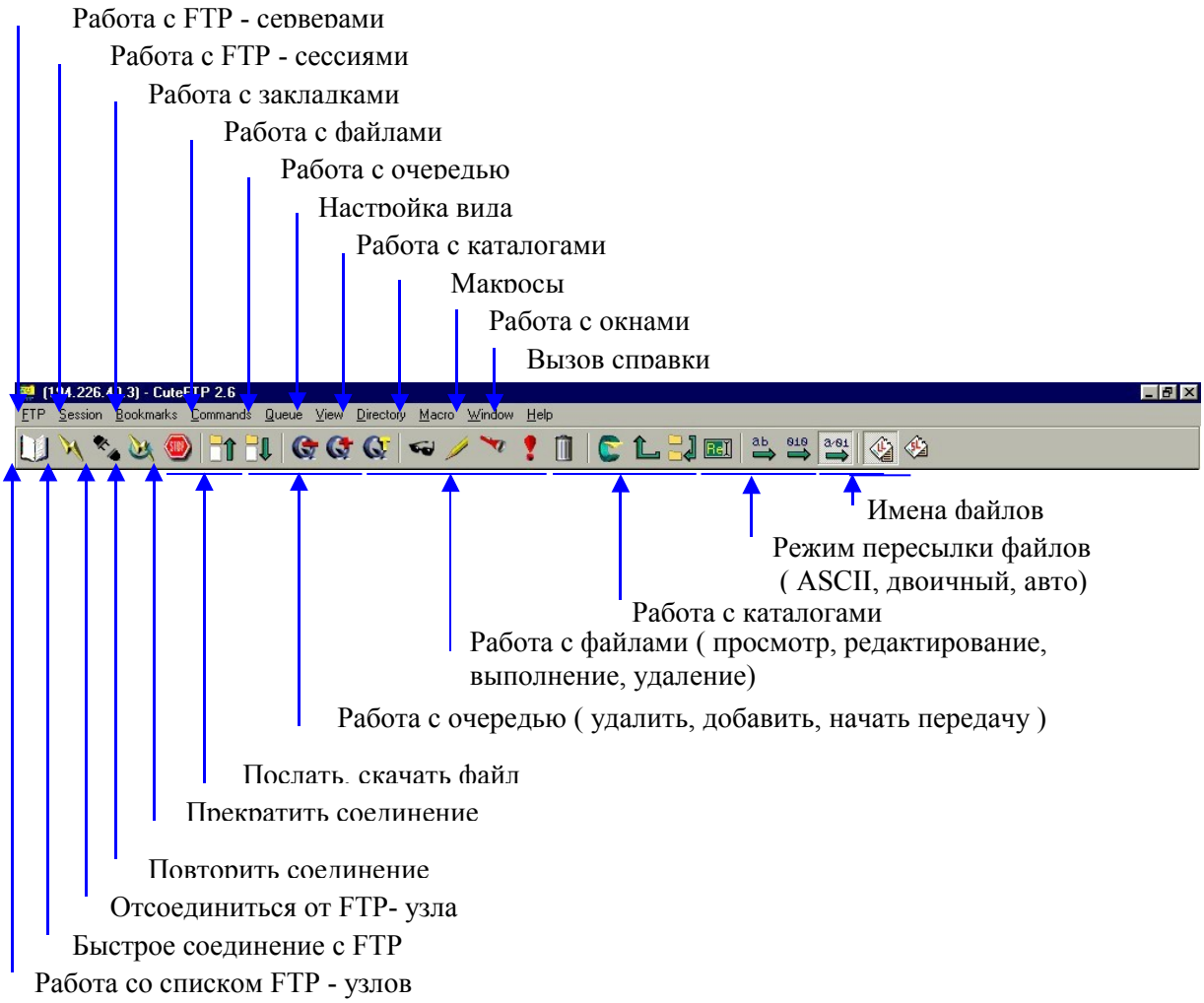

Рисунок 2.2 – Меню программы

На рисунке 2.3 приведено окно *Site Manager*

Рисунок 2.3- Вид окна *Site Manager*

Site Manager позволяет создавать закладки на FTP - узлы, тем самым облегчая работу с большим количеством FTP - узлов в сети Internet.

На рисунке 2.4 приведен пример окна команд сервиса FTP и коды ответов на эти команды (таблица 2.1).

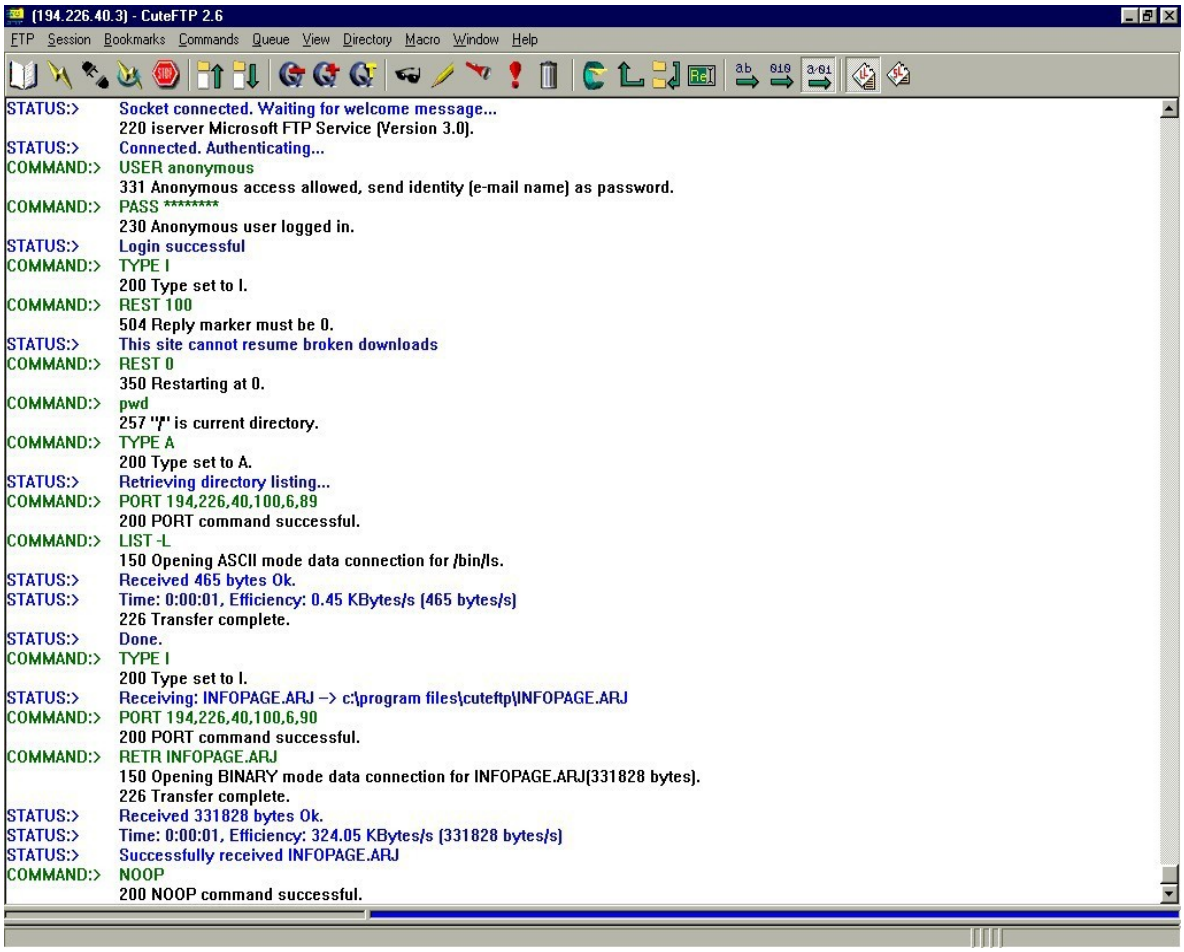

Рисунок 2.4- Окно команд сервиса FTP

Таблица 2.1- Коды ответов FTP-сервера

Код	**Описание**
110	Маркер рестарта
120	Служба будет готова через n минут
125	Соединение данных уже открыто, передача начата
130	Файл доступен, открывается соединение данных
200	Команда выполнена
202	Команда не реализована
211	Состояние системы или системная подсказка
212	Состояние каталога
213	Состояние файла
214	Сообщение-подсказка

215	Тип системы NAME
220	Сервис готов для нового пользователя
221	Служба закрывает управляющее соединение. Если необходимо, производится выход из системы
225	Соединение данных открыто, передача отсутствует
226	Закрытие соединения данных, требуемая операция выполнена успешно
227	Начало пассивного режима
230	Пользователь зарегистрирован в системе, можно работать
250	Требуемая операция с файлом выполнена успешно
257	Маршрут PATHNAME создан
331	Имя пользователя в порядке, требуется пароль
332	Для доступа к системе требуется указывать ресурс
350	Для операции с файлом необходима дополнительная операция
421	Служба недоступна, управляющее соединение закрывается
425	Сервер не может открыть соединение данных
426	Соединение закрыто, передача прервана
450	Требуемая операция не принята. Файл недоступен
451	Операция по передаче прервана: местная ошибка при обработке
452	Требуемая операция не принята. Недостаточно места для размещения файла в системе
500	Синтаксическая ошибка в параметрах или аргументах
502	Команда не реализована
503	Неправильная последовательность команд
504	Команда не реализована для этого параметра
530	Не произведена регистрация в системе
532	Требуется ресурс для хранения файлов
550	Требуемая операция не принята. Файл недоступен
551	Требуемая операция не принята. Тип страницы неизвестен
552	Операция по передаче файла прервана. Превышение зарезервированного для хранения места
553	Требуемая операция не принята. Имя файла не разрешено в системе

2.3.2. Программы PING и TRACEROUTE

Traceroute - это трассировка пути, то есть прослеживание пути следования IP- пакетов от вашего компьютера до любого другого. Программа tracert.exe выдает на экран имена или, в крайнем случае, IP-адреса компьютеров, через которые проходят IP-пакеты от вашего компьютера до

тестируемого. Программа посылает на все передающие станции по 3 пакета и выводит на экран время связи между ними.

Pinging - это передача тестовой информации на компьютер по его известному IP-адресу для определения времени передачи данных. Программа Ping.exe, посылающая тестовые данные, является основным инструментом для проверки работоспособности вашего компьютера, надежности связи вашего провайдера с крупными каналами связи и каких-то web-сайтов.

Выйдите в NC (или "Пуск/Выполнить") и наберите команду "ping 193.0.0.193" (этот адрес должен тестироваться в любое время). Вы увидите несколько примерно таких строчек:"193.0.0.193…2ms", где для Вас важно время (2ms) - это время, за которое информация с вашего компьютера передается на тестируемый компьютер 193.0.0.193. Если же в ответ увидите лишь сообщения об ошибках, то связи между вашими компьютерами нет. В большинстве случаев тестировать соединение можно как по IP-адресу, так и по DNS, например, "ping 195.1.2.99" или "ping conter.download.ru".

2.4. Вопросы к защите лабораторной работы

1. Назовите основные этапы работы в FTP.
2. Какие команды FTP используются для передачи файла?
3. Назовите опции FTP для операций по передаче файла.
4. Что означает тип файла в FTP? Что такое формат файла в FTP?
5. Как FTP использует протокол TELNET на управляющем соединении?
6. Каково соотношение FTP и TELNET ?
7. Как протестировать соединение?

2.5. Литература

1. Романчева Н.И., Соломенцев В.В. Информационные ресурсы Интернет. / Учебное пособие. –М.: МГТУ ГА, 2000.

2. Джамса К., Коуп К. Программирование для Internet в среде Windows.- СПб: Питер, 1996.

3 ЛАБОРАТОРНАЯ РАБОТА №3
ОСНОВЫ РАБОТЫ С PHP- МАШИНОЙ. СОЗДАНИЕ СЦЕНАРИЯ И АДМИНИСТРИРОВАНИЕ КОНФЕРЕНЦИИ

3.1 Цель лабораторной работы

Получение навыков работы с PHP-машиной, создание и администрирование сценария конференции.

3.2 Задание на выполнение лабораторной работы

1) Запустить PHP-сценарий для вывода информации о версии PHP-машины.
2) Разработать программу и сценарий для решения уравнения (в соответствии с вариантом, выданным преподавателем). Предусмотреть формы ввода данных (однострочные поля) и кнопки управления (например: «Решить», «Очистить»).
3) Разработать программу для ввода и получения данных пользователем. Документ ввода данных должен содержать 2 поля для однострочного ввода: «Имя пользователя», «e-mail», поле многострочного ввода, например «Комментарий», переключатели единственного и множественного выбора. Если пользователь не сделал выбор, считать выбором установленный по умолчанию.
4) Выполнить работу с файлами, используя функции empty () и file().
5) Проверить открытие файла PHP-сценарием.
6) Разработать программу и сценарий простой конференции в сети. Название темы должно задаваться администратором сайта (текстовый файл). В случае отсутствия тем должно выводится соответствующее сообщение. Посетители могут оставить свое мнение в текстовом формате и прочитать мнения, оставленные посетителями ранее. Выполнить анализ введенного имени пользователя, в том числе используя функцию htmlspecialchars ().В

HTML- документе использовать табличный способ форматирования информации. Создать файл tema.txt, в который будет записываться текст новой темы для обсуждения.

7) Выполнить администрирование конференции с помощью FTP-клиента.

в файл nN.txt записывать количество сообщений по теме с номером N,

в файл mN.txt записывать сообщения по теме с номером N.

В качестве атрибута формы action в файле index.php указать файл add.php.

3.3 Основные теоретические сведения

3.3.1 Основы PHP

PHP (Hypertext Preprocessor) – препроцессор гипертекста - является языком серверных сценариев для быстрого создания динамических web-страниц, его поддерживают практически все web-серверы, вне зависимости от их платформ. Для написания PHP-сценариев можно использовать любой редактор, при этом следует дать файлу расширение .php.

Основное преимущество PHP- простота, гибкость, скорость выполнения.

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

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

1) ***<? echo (“SGML инструкции | n”) ; ?>***

2) ***<?php echo (“XML документ | n”) ; ?>***

3) ***<script language= “php”>***

echo (“специально для FrontPage”);

</script>

4) ***<% echo (“ASP-стиль”) ; %>***

<% = $variable; # комментарий “echo (“ASP-стиль”) ; %>

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

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

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

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

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

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

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

Если требуется проверка переменной на соответствие определенному типу данных, то применяются функции *is_integer(), is_double(), is_string(), is_(array), is_object( )* и *is_bool()*. В случае соответствия типа переменной

выводится 1, иначе- 0.

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

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

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

В PHP заранее определен ряд констант, которые называются предопределенными. Их основное назначение – хранить информацию о системе. Например, PHP_VERSION, PHP_OS содержат соответственно версию PHP и название операционной системы, на которую установлен сервер.

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

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

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

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

?>

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

<?php

echo $_GET [‘text’] ; // выводит значение поля text

?>

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

Несмотря на то, что базы данных являются более удобными хранилищами данных, файлы до сих пор не утратили своей актуальности. При малых количествах информации (счетчик посещений, гостевая книга) лучше использовать именно файлы.

Для работы с файлом его нужно открыть. Для этого используется функция *fopen ()*. В качестве входных параметров выступает строка с именем файла и специальный признак, по которому определяется режим открытия файла. Функция возвращает дескриптор файла, который имеет значение типа source, в случае неудачной попытки открытия файла она возвратит FALSE. Пример PHP-кода:

<?php

$file = fopen (“info.txt”, “r”); // открытие файла info.txt для чтения

?>

Режимы открытия файла приведены ниже:

r - файл открывается только для чтения. При попытке открыть несуществующий файл выводится сообщение об ошибке. Если файл открылся, то указатель текущей позиции устанавливается в начало;

r+ - файл открывается одновременно для чтения и записи. Указатель

текущей позиции устанавливается в начало файла. При этом запись в файл будет происходить поверх уже существующих данных;

w - файл открывается для записи. При этом все данные, которые были в нем, уничтожаются. Если файл с таким именем не существует, то он создается;

w+ - действия аналогичные режиму w, но файл открывается для чтения и для записи;

a - файл открывается для записи. При этом его указатель устанавливается в конец файла. Если файла не существует, то выводится сообщение об ошибке;

a+ - файл открывается для чтения и записи. При этом его указатель устанавливается в конец файла. Если файла не существует, то он создается.

Если работа с файлом завершена, то его следует закрыть. Эта операция осуществляется с помощью функции *fclose (),* которая в качестве входного параметра принимает дескриптор файла. Функция возвращает булевское значение, которое равно TRUE при удачном закрытии файла и FALSE в противном случае. Пример PHP-кода:

```
<?php
$file = fopen (“info.txt”, “r”);  // открытие файла только для чтения
//закрытие файла
if  (fclose ($file))
{
  echo “закрытие файла произошло удачно”;
}
else
{
  echo “ошибка закрытия файла”;
}
?>
```

Чтение из файлов в PHP можно осуществлять различными способами.

Например, с помощью функции *fread()*. Она принимает дескриптор файла – число, которое определяет длину строки из файла и возвращает эту строку. В приведенном ниже примере из файла info.txt читаются первые пять символов, которые записываются в виде строки в переменную $txt:

```
<?php
$file  =  fopen (“info.txt”, “r”);   // открытие файла только для чтения
// чтение 5 символов файла
$txt = fread ($file, 5  );
//закрытие файла
fclose ($file);
//вывод строки
echo  $txt;
?>
```

Если при чтении достигнут конец файла, функция прекращает свое выполнение и возвращает строку, содержащую ровно столько символов, сколько прочиталось (несмотря на то, что длина строки может быть меньше запрашиваемой).

В PHP имеется похожая на выполнение функция *fgets(),* которая читает символы из файлов в количестве равном заданному минус единица и возвращает их. Чтение прекращается не только при достижении конца файла, но и если встречается символ перевода строки \n. Эту функцию удобно применять в том случае, когда нужно просмотреть файл по строкам:

```
<?php
$file  =  fopen (“info.txt”, “r”);   // открытие файла только для чтения
// чтение 4-х символов файла
$txt = fgets ($file, 5  );
fclose ($file);
//вывод строки
echo  $txt;
?>
```

3.3. 2 Примеры сценариев

Для того чтобы узнать версии PHP- машины, установленной на вашем компьютере или на сервере, услугами которого Вы пользуетесь, необходимо запустить сценарий, приведенный ниже:

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

На примере решения квадратного уравнения можно ознакомиться с основными принципами построения PHP- программ. В приведенных примерах сценариев пронумерованы строки, которые относятся к PHP-коду. Непронумерованные строки представляют собой простой HTML-код.

```
<html><head><title>Квадратное уравнение</title></head><body>
1<?php
2       if (isset($_POST['act'])&& $_POST['act'] == 'act') {
3                       if(((!is_numeric($_POST['a']))||(!is_numeric($_POST['b'])))||(!
is_numeric($_POST['c']))){
4 exit("Ошибка ввода!<br><br><a href=". $_SERVER['PHP_SELF'] .">Ешё
раз</a>");}
5   echo"a = ".$_POST['a'];$a = $_POST['a'];
6   echo"  b = ".$_POST['b'];$b = $_POST['b'];
7   echo"  c = ".$_POST['c'];$c = $_POST['c'];
8   $d =$b*$b-4*$a*$c;echo"  d = ", $d;
9 if($d<0)echo "<br>Нет вещественных корней!";
10 if($d>0)echo  "<br>x1  =  ",((-$b+sqrt($d))/(2*$a)),"  x2  =  ",((-$b-sqrt($d))/(2*
$a));
11 if($d==0)echo"<br>x1 = x2 = ",((-$b)/(2*$a));
12 echo"<br><br><a href=". $_SERVER['PHP_SELF'] .">Следующее</a>";
exit;}
13 else{
14 ?>
```

15 <form action="<?php echo $_SERVER['PHP_SELF'];?>" method="POST">

*Введите коэффициенты квадратного уравнения a, b, c

*

a = <input type="text" name="a">

b = <input type="text" name="b">

c = <input type="text" name="c">

*<input type="hidden" name="act" value="act">

 //скрытая переменная*

<input type="submit" name="submit" value="Решить">

<input type="reset" name="reset" value="Очистить">

</form></body></html>

16 <?php

17 }

18 ?>

1-я и 16-я строки содержат символы начала PHP-кода, а 14-я и 18-я — символы конца PHP-кода соответственно. В15-й строке - внедрение PHP-кода между открывающим и закрывающим символами HTML-кода. Атрибут action указывает путь к ресурсу, на который возложена задача обработки введенных пользователем данных. В данном случае сам этот файл и является тем ресурсом, который обрабатывает данные, введенные пользователем (можно было бы записать action="kvadrat.php". Во 2-й строке написано: если (if) некоторое условие выполняется, то выполняется код с 3-й по 12-ю строку, а иначе (else) переходим к 14-й строке и выполняем дальше по порядку.

Допустим, что условие в 3-ей строке не выполнено, тогда интерпретатор в 14-й строке закроет PHP-код, в 15-й строке выдаст название файла и «отдохнет» до 16-й строки. В 16-й строке он опять включится, выдаст закрывающую операторную скобку и отключится опять, закрыв PHP-код в 18-й строке. Строки, находящиеся между 14-й и 16-ой будут переданы веб-сервером клиентскому браузеру, который выведет на экран форму ввода данных для пользователя.

В строке *// скрытая переменная* в форму вводится невидимая переменная, которая будет играть важную роль в PHP-коде. Тип этой переменной *hidden* можно перевести на русский язык как «спрятанная» или «скрытая», то есть переменная спрятана от пользователя. До щелчка на кнопке Submit переменные никуда не передаются. После щелчка на этой кнопке ресурс, указанный в атрибуте формы *action,* начинает их анализировать. Для передачи данных применен метод POST. При использовании этого метода передаваемые переменные заполняют массив *$_POST['имя переменной]* значениями соответствующей переменной *$_POST[имя переменной]=значение переменной*.

Во 2-й строке проверяется сложное условие, состоящее из двух частей. Первая часть условия - установлена ли *(isset)* переменная *act*. Второе условие - равно ли *(==)* значение этой переменной, в данном случае *act*. Эти условия соединены знаком &&, который означает, что они должны выполняться оба одновременно. Это условие, если нет неисправностей на сервере, всегда выполняется при щелчке на кнопке *Submit.* А если сложное условие выполнено, PHP-интерпретатор выполняет строки кода с 2-ой по 12-ю. 5-7 строки одинаковы, они выводят на экран пользователя введенные им ранее значения. То, что заключено в двойные кавычки, функция echo отображает на экране без изменений, как правило, отображая и пробелы (иногда не все). Вместо символов элементов массива отображаются их значения. Для соединения подписи со значением служит точка (можно было использовать и запятую). Оператор вывода на экран echo завершается точкой с запятой. Следующий в каждой строке - оператор присвоения. Значения элементов массива присваиваются трем переменным — $a, $в и $c соответственно. В 8-й строке, используя уже введенные переменные, определяется дискриминант $d и выводится его значение на экран с помощью оператора echo. Для разнообразия используем в качестве связки между надписью и значением не точку, как в предыдущих строчках, а запятую.

Из математики известно, что в зависимости от значения дискриминанта $d существует три варианта решения квадратного уравнения. В зависимости от

значения $d выполняется одна из строк кода - 9я, 10-я или 11-я. Все три строки начинаются с проверки условия, и если условие не выполняется, то строка пропускается PHP-интерпретатором.

В программе необходимо тщательно анализировать вводимую информацию. В строках 3, 4 используются два логических оператора «!» - НЕ и «||» - ИЛИ, а также встроенная PHP-функция *is_numeric(значение).* Эта функция является истинной (true), если значение введено и является числом. В противном случае функция возвращает ложь (false). Условие в строке 3 можно объяснить так: если хотя бы одно из введенных пользователем значений отсутствует или не является числом, следует выполнять код строки 4, иначе строка 4 пропускается. 4-я строка очень похожа на 12-ю, однако вместо функции echo используется функция exit (информация). А это означает, что после вывода информации из круглых скобок функции exit() программа завершается. При этом пользователь получит сообщение об ошибке и возможность вернуться к началу.

В предыдущем сценарии использовались однострочные поля ввода. Однако в сети встречаются и другие способы ввода информации пользователем. Пример сценария с использованием множественного и единственного выбора представлен ниже.

```
<html><head><title>Data Form</title></head><body>
<?php
if (isset($_POST['act'])&& $_POST['act'] == 'act') {
        echo"ИМЯ ".$_POST['name'];
        echo"<br>Email ".$_POST['email'];
        echo"<br><u>Выбрано элементов - ".(count($_POST) - 6)."</u>";
   for($j=0;$j<5;$j++){
   if(isset($_POST['chk'.$j]))echo"<br>Выбрано - ".$_POST['chk'.$j];
   }
         echo"<br>Единственный выбор -  ".$_POST['rad'];
         echo"<br>Комментарий - ".$_POST['coment'];
        echo"<br><br><a href=". $_SERVER['PHP_SELF'] .">Еще раз</a>";
```

```
        exit;}
 else{
?>
<form action="<?php echo $_SERVER['PHP_SELF'];?>" method="POST">
  Имя:  <input type="text" name="name"><br>
   Email:<input type="text" name="email"><br><br>
   Множественный выбор:<br>
<?php
for($i=1; $i<4; $i++){
echo "<INPUT TYPE='checkbox' NAME='chk$i' VALUE='Выбор $i'>Выбор $i<br>";
}
?>
<br>Единственный выбор:<br>
<INPUT TYPE=radio NAME="rad" VALUE="Выбор 1" CHECKED>Выбор 1<br>
<INPUT TYPE=radio NAME="rad" VALUE=" Выбор 2">Выбор 2<br>
<INPUT TYPE=radio NAME="rad" VALUE="Выбор 3">Выбор 3<br><br>
Комментарий:<br>
<textarea name="coment" rows=5 cols=20 >Здесь комментарий
</textarea><br><br>
<input type="hidden" name="act"  value="act">
<input type="submit" name="submit" value="Отправить">
<input type="reset" name="reset" value="Очистить">
</form></body></html>
<?php
}
?>
```

Под конференцией понимается программа, позволяющая любому посетителю оставить свое мнение по теме, заданной администратором сайта, а также возможность прочитать мнения, оставленные посетителями ранее. Сценарий создаваемой в данной лабораторной работе конференции состоит из 3-х файлов, четвертый текстовый будет формироваться динамически. Листинг файла index.php представлен ниже.

```
<html><head><title>Конференция</title></head><body><center>
<table height="100%" width="100%"><tr><td align=center>
<table BGCOLOR='b0e0e6' width='90%'><tr>
<td width="15%"><a href='#'>Ссылка</a></td>
<td width="70%" align=center>
<font face='verdana' color='#0000ff'><b>КОНФЕРЕНЦ-ЗАЛ</td>
<td align=right><a href='#'>К сайту</a></td></tr></table>
<table BGCOLOR='b0e0e6' width='90%'><tr><td align=center><br>
<?php
$m = @file("tema.txt");
if(!$m)exit("НЕТ ТЕМ!");
echo "<b><font color='#ff0000'>".$m[0];
?>
</td></tr></table>
<table BGCOLOR="b0e0e6" width="90%" CELLPADDING=20><tr><td>
<?php
if (file_exists("m.txt")){
$t = file("m.txt");
}else{
$fp = @fopen("m.txt","w");
fclose($fp);
}
if (!empty($t)){
?>
<center><TEXTAREA ROWS=15 COLS=70>
<?php
for($i=0;$i<count($t);$i++){
echo $t[$i];
}
}
?>
</TEXTAREA>
```

```
</td></tr></table>
<H5><font face="verdana" color="#0000ff">Ниже можно высказаться</h5>
<table BGCOLOR="b0e0e6" width="90%">
<form action="add.php" method="POST">
<tr><td align=center><br>
<font color="#0000ff">Имя (до 30 зн.)
<input name="nam" type="text" SIZE=25 MAXLENGTH=35></td>
<td align=center><font color="#0000ff">
Email <input name="mail" type="text">
</td></tr></table>
<table BGCOLOR="b0e0e6" width="90%"><tr>
<td align=center>
<table width="100%"><tr>
<td align=center>
<font color="#0000ff">Ваше мнение (до 900 зн.)<br>
<textarea name="mes" rows=5 cols=40 ></textarea>
</td></tr></table>
</td><tr><td align=center><br>
<input type="reset" value="Очистить">
<input type="submit" value="Отправить">
</form>
</td></tr></table>
</td></tr></table></body></html>
```

Листинг файла add.php :

```
<html><title>Добавление мнения</title><body>
<?php
if(!empty($_POST['nam'])){
$tn=htmlspecialchars($_POST['nam'],ENT_QUOTES);
}else{
$tn='MisterX';
}
```

```
if(!empty($_POST['mail'])){
$te=htmlspecialchars($_POST['mail'],ENT_QUOTES);
}else{
$te='не введен';
}
if ((strlen(rtrim($_POST['mes'])))!=0){ //проверяем пустую строку
$t2=htmlspecialchars($_POST['mes'],ENT_QUOTES);
}
else{
exit("<table cellspacing='5' cellpadding='5' width='100%' height='100%'>
<tr><td><center><H3>Вы не ввели сообщение.</h3>
<br><INPUT TYPE='button' VALUE='НАЗАД' onClick='history.go(-1)'>
</td></tr></table>");}
$dat = date("d m y H:i");
$fp = @fopen ("m.txt", "a");
if(!$fp)exit("<center>Не могу открыть файл <b>m.txt</b>");
fwrite ($fp, $dat." пишет ".$tn." Email ".$te."\n".$t2."\n\n");
fclose ($fp);
exit("<table cellspacing='5' cellpadding='5' width='100%' height='100%'>
<tr><td><center><H3>Ваше сообщение успешно загружено.</h3>
<br><a href='index.php'><H4>НА КОНФЕРЕНЦИЮ</h4></a>
</td></tr></table></body></html>");
?>
```

Результат выполнения файла представлен на рисунке 3.1

3.4 Вопросы к защите лабораторной работы

1) Что такое PHP, его возможности?

2) Поясните способы отделения PHP-кода от общего кода HTML.

3) Сформулируйте правила для выбора имени переменной в PHP.

4) Поясните способы ввода данных пользователем.

5) Поясните алгоритм проверки открытия файла PHP-сценарием.

6) Что понимается под конференцией?

7) Как выполняется проверка вводимой пользователем информации?

8) Перечислите базовые функции PHP и их синтаксис.

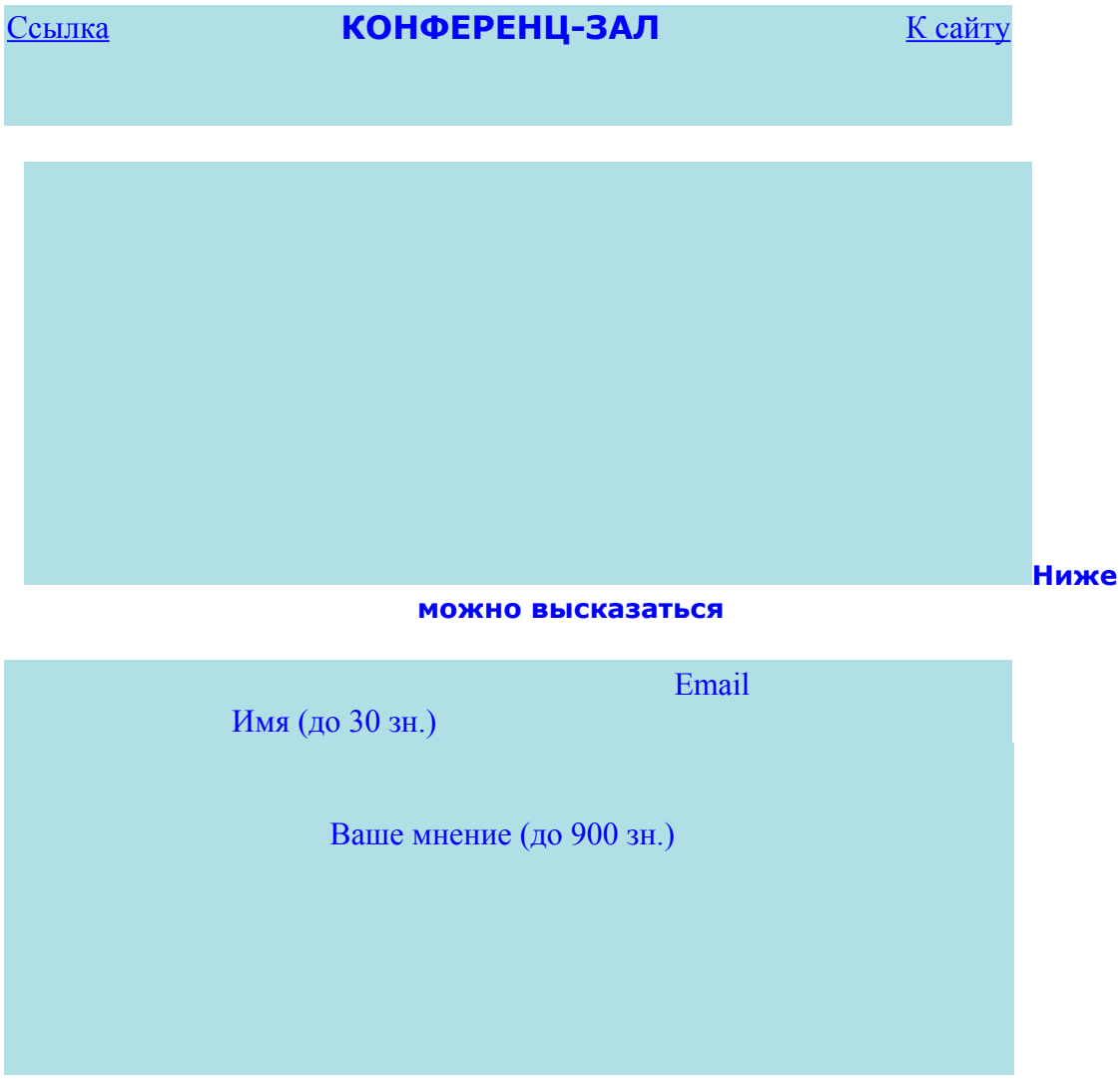

Рисунок 3.1 – Вид главной страницы конференции

3.5 Литература

1) Мазуркевич А., Еловой Д. PHP: настольная книга программиста - Мн.: Новое знание, 2003. - 480 с.

2) Котеров Д.В. Самоучитель PHP 4.- СПб.: БХВ-Петербург, 2001. - 576 с.

3) http://microsoft.com/php