

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

Федеральное агентство образования

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
“ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

**Л.Ф. ГАЙСИНА**

**FTP-КЛИЕНТ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ**

Рекомендовано к изданию Редакционно-издательским советом  
государственного образовательного учреждения  
высшего профессионального образования  
“Оренбургский государственный университет”

Оренбург 2005

ББК 32.973.202я7  
Г 14  
УДК 004.732 (07)

Рецензент доктор технических наук, профессор Соловьев Н.А.

**Гайсина Л.Ф.**

**Г – 14   FTP-клиент: Методические указания к лабораторному практикуму. - Оренбург: ГОУ ОГУ, 2005. – 35 с.**

Методические указания посвящены изучению протокола прикладного уровня стека TCP/IP – протокола пересылки файлов ftp. Работа этого протокола основана на архитектуре “клиент-сервер”. В методических указаниях представлены материалы, необходимые для подготовки студентов к написанию программы ftp-клиент. Данная программа осуществляет подключение к ftp-серверу, закачивает файлы на ftp-сервер, скачивает файлы с ftp-сервера, просматривает файлы, каталоги, удаляет файлы, каталоги.

В описание лабораторной работы включены цель работы, порядок ее выполнения, рассмотрены теоретические вопросы, связанные с реализацией поставленной цели, приведены контрольные вопросы, тесты для самопроверки, указан список использованных источников. В приложениях представлены правила оформления результатов лабораторной работы.

Методические указания предназначены для выполнения лабораторного практикума по дисциплине “Сети ЭВМ и телекоммуникации” для студентов специальности 220400 – “Программное обеспечение вычислительной техники и автоматизированных систем”, 220100 – “Вычислительные машины, комплексы, системы и сети” в рамках направления подготовки дипломированного специалиста 654600 – “Информатика и вычислительная техника”.

ББК 32.973.202я7

© Гайсина Л.Ф., 2005  
© ГОУ ОГУ, 2005

## Введение

Сложный характер и динамизм современных мирохозяйственных связей привел к необходимости создания новых телекоммуникационных технологий, порождающих новые услуги и соответственно потребность в них.

Объем и способы информирования специалистов с помощью средств компьютерных коммуникаций коренным образом изменились за последние год-два. И если ранее подобные средства предназначались лишь для узкого круга специалистов и опытных пользователей, то теперь они рассчитаны на самую широкую аудиторию.

В настоящее время передача данных с помощью компьютеров, использование локальных и глобальных компьютерных сетей становится столь же распространенным, как и сами компьютеры.

Целью данных методических указаний является обучение студентов работе с протоколом ftp на основе архитектуры “клиент-сервер”. В качестве транспорта протокол ftp использует протокол TCP/IP

По окончании изучения представленных методических указаний студенты будут уметь писать программу ftp-клиент, которая осуществляет подключение к ftp-серверу, закидывает файлы на ftp-сервер, скачивает файлы с ftp-сервера, просматривает файлы, каталоги, удаляет файлы, каталоги.

# **1. Лабораторная работа – FTP-клиент**

## **1.1 Цель работы**

Ознакомиться с принципами организации архитектуры “клиент-сервер”, основными ее достоинствами и недостатками. Изучить стек TCP/IP, основные уровни, протоколы. Познакомиться с протоколом прикладного уровня FTP (File Transfer Protocol). Научиться написанию программ с использованием протокола FTP (программа FTP-клиент).

## **1.2 Постановка задачи**

Написать программу ftp-клиент, которая осуществляет подключение к ftp-серверу, закачивает файлы на ftp-сервер, просматривает файлы, каталоги, удаляет файлы, каталоги.

## **1.3 Методические указания**

### **1.3.1 Архитектура "клиент-сервер"**

Представим, что существует машина А и на ней выполняется приложение А и существует машина В. Любая программа, которой потребуется печать на "чужом" принтере, должна включать в себя функции, подобные тем, которые выполняет приложение А. Но нагружать этими стандартными действиями каждое приложение — текстовые и графические редакторы, системы управления базами данных и другие приложения — не очень рационально.

Гораздо выгоднее создать специальный программный модуль, который будет выполнять формирование сообщений-запросов к удаленной машине и прием результатов для всех приложений. Такой служебный модуль называется клиентом.

На стороне же компьютера В должна работать другая специализированная программа — сервер, постоянно ожидающий прихода запросов на удаленный доступ к ресурсу (принтеру или файлам, расположенным на диске) этого компьютера. Сервер, приняв запрос из сети, обращается к локальному периферийному устройству с участием локальной ОС.

На сегодняшний день появилась возможность иметь вычислительные и информационные ресурсы на рабочем столе пользователя и управлять ими по собственному желанию с помощью графического интерфейса. Увеличение производительности персональных компьютеров позволило перенести части системы (интерфейс с пользователем, прикладную логику) для выполнения непосредственно на рабочем месте, а функции обработки данных оставить на центральном компьютере. Таким образом, система стала распределенной - одна часть функций выполняется на центральном компьютере, другая - на

персональном, который связан с центральным посредством коммуникационной сети. Появилась клиент-серверная модель взаимодействия компьютеров и программ в сети и на этой основе стали развиваться средства разработки приложений для реализации информационных систем.

Очень удобной и полезной функцией клиентской программы является способность отличить запрос к удаленному файлу от запроса к локальному файлу. Если клиентская программа умеет это делать, она сама распознает и перенаправляет (redirect) запрос к удаленной машине. Отсюда и название, часто используемое для клиентской части — редиректор. Иногда функции распознавания выделяются в особый программный модуль, в этом случае редиректором называют не всю клиентскую часть, а только этот модуль.

Программные клиент и сервер выполняют системные функции по обслуживанию запросов всех приложений компьютера А на удаленный доступ к файлам компьютера В. Чтобы приложения компьютера В могли пользоваться файлами компьютера А, описанную схему нужно симметрично дополнить клиентом для компьютера В и сервером для компьютера А. Схема взаимодействия клиента и сервера с приложениями и локальной операционной системой приведена на рисунок 1.

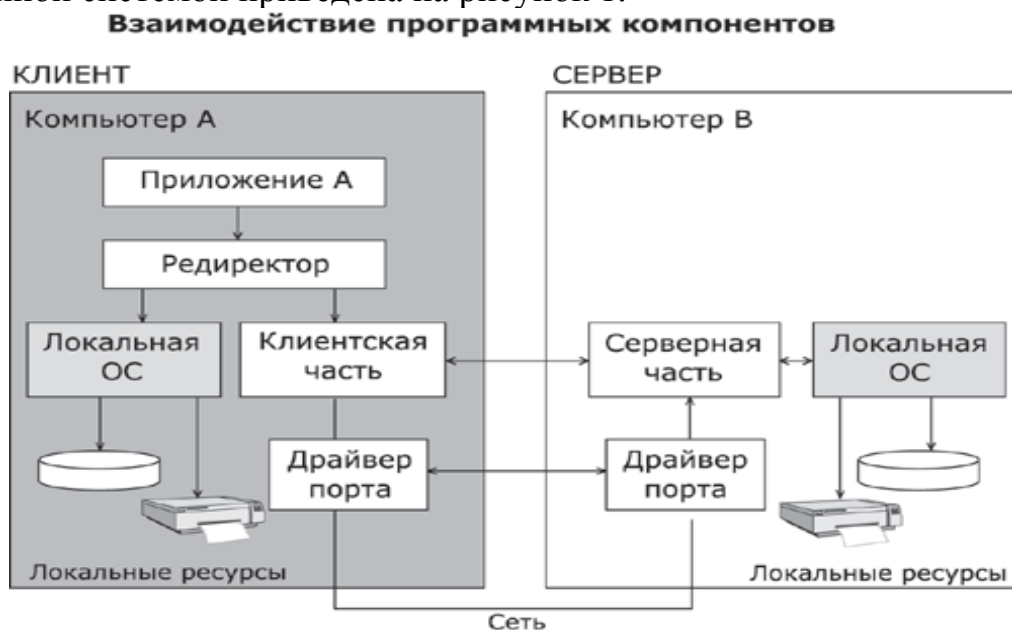


Рисунок 1 - Взаимодействие программных компонентов при связи двух компьютеров

В последнее время долю обработки, приходящуюся на клиента, стали называть "толщиной" клиента.

Развитие архитектуры "клиент-сервер" происходит по спирали и в настоящее время намечается тенденция централизации вычислений, то есть замены "толстых" клиентов - рабочих станций на основе высокопроизводительных ПЭВМ, оснащенных мощным программным обеспечением для поддержки прикладных программ, мультимедийных средств, навигационного и графического интерфейса - "тонкими" клиентами.

Характерный пример "тонкого" клиента - архитектура Sun Ray Hot Desk, предложенная компанией Sun Microsystems.

Архитектура Sun Ray Hot Desk предполагает использование настольных систем, имеющих минимум программных и аппаратных средств, но обладающих широкими возможностями работы с приложениями в соответствии с основной идеей "тонких" клиентов - вынести на сервер все, вплоть до виртуальных драйверов устройств, включая драйвер монитора (рисунок 2).



Рисунок 2 - Ранжирование клиентов по "толщине"

Компоненты "клиент" и "сервер" не обязательно должны работать на разных машинах, хотя обычно это так и есть – клиентское приложение находится на рабочей станции пользователя, а сервер - на специальной выделенной машине. Наиболее распространены следующие виды серверов: файл-серверы, серверы баз данных, серверы печати, серверы электронной почты, Web-сервер и другие.

Однако двухуровневая архитектура "клиент-сервер" имеет также недостатки, такие, например, как сложность администрирования и низкая информационная безопасность.

### 1.3.2 Стек TCP/IP

Главная цель, которая преследуется при соединении компьютеров в сеть - это возможность использования ресурсов каждого компьютера всеми пользователями сети. Для того, чтобы реализовать эту возможность, компьютеры, подсоединенные к сети, должны иметь необходимые для этого средства взаимодействия с другими компьютерами сети. Задача разделения сетевых ресурсов является сложной.

Обычным подходом при решении сложной проблемы является ее декомпозиция на несколько частных проблем - подзадач. Для решения

каждой подзадачи назначается некоторый модуль. При этом четко определяются функции каждого модуля и правила их взаимодействия.

Частным случаем декомпозиции задачи является многоуровневое представление, при котором все множество модулей, решающих подзадачи, разбивается на иерархически упорядоченные группы - уровни. Для каждого уровня определяется набор функций-запросов, с которыми к модулям данного уровня могут обращаться модули выше лежащего уровня для решения своих задач. Такой формально определенный набор функций, выполняемых данным уровнем для выше лежащего уровня, а также форматы сообщений, которыми обмениваются два соседних уровня в ходе своего взаимодействия, называется *интерфейсом*.

Интерфейс определяет совокупный сервис, предоставляемый данным уровнем выше лежащему уровню.

Согласованный набор протоколов разных уровней, достаточный для организации межсетевого взаимодействия, называется *стеком протоколов*.

Существует достаточно много стеков протоколов, широко применяемых в сетях. Это и стеки, являющиеся международными и национальными стандартами, и фирменные стеки, получившие распространение благодаря распространенности оборудования той или иной фирмы. Примерами популярных стеков протоколов могут служить стек IPX/SPX фирмы Novell, стек TCP/IP, используемый в сети Internet и во многих сетях на основе операционной системы UNIX, стек OSI международной организации по стандартизации, стек DECnet корпорации Digital Equipment и некоторые другие.

**Стек TCP/IP**, называемый также стеком Internet, является одним из наиболее популярных и перспективных стеков коммуникационных протоколов. Так как стек TCP/IP был разработан до появления модели взаимодействия открытых систем ISO/OSI, соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.



Рисунок 3 – Соответствие стека протоколов TCP/IP модели OSI

Как и в модели OSI, данные более верхних уровней инкапсулируются в пакеты нижних уровней (рисунок 4).

Стек протоколов TCP/IP делится на 4 уровня: *прикладной (application)*, *транспортный (transport)*, *межсетевой (internet)* и *уровень доступа к среде передачи (network access)*. Структура протоколов TCP/IP приведена на рисунке 5.

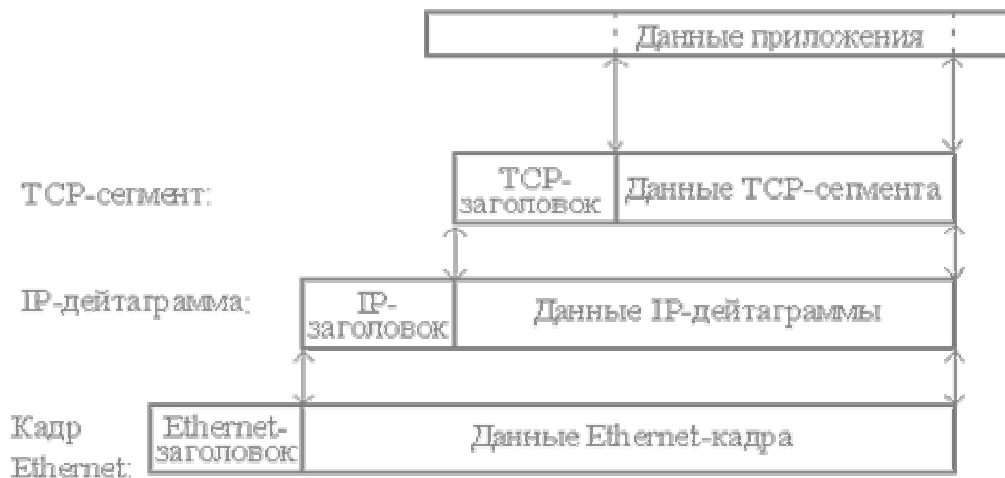


Рисунок 4 - Пример инкапсуляции пакетов в стеке TCP/IP

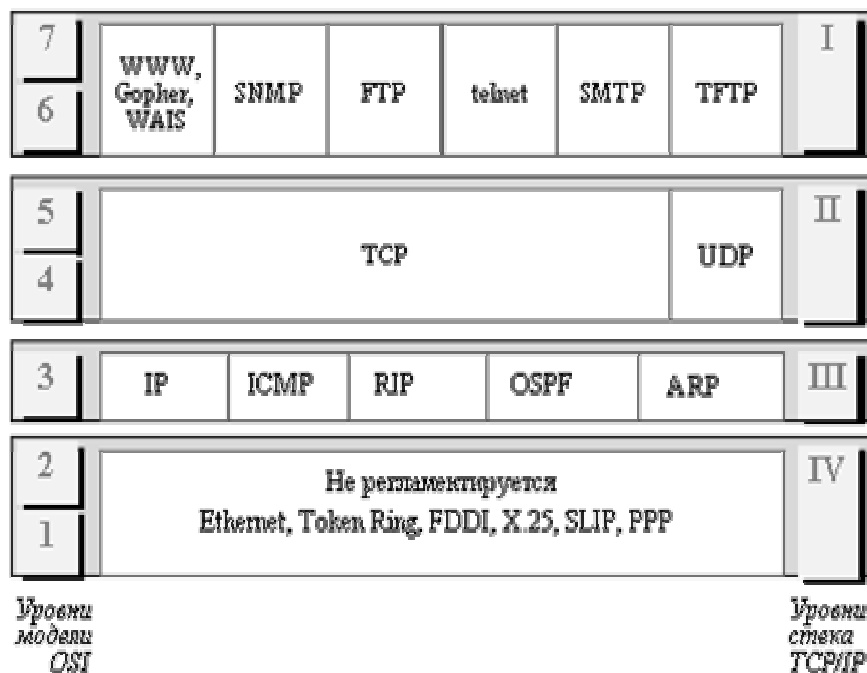


Рисунок 5 - Стек TCP / IP

Ниже кратко рассматриваются функции каждого уровня и примеры протоколов.

Самый нижний (уровень IV) - уровень доступа к среде передачи - соответствует физическому и каналному уровням модели OSI. Этот уровень в протоколах TCP/IP не регламентируется, но поддерживает все популярные стандарты физического и канального уровня: для локальных каналов это Ethernet, Token Ring, FDDI, для глобальных каналов - собственные протоколы работы на аналоговых коммутируемых и выделенных линиях



SLIP/PPP, которые устанавливают соединения типа "точка - точка" через последовательные каналы глобальных сетей, и протоколы территориальных сетей X.25 и ISDN. Разработана также специальная спецификация, определяющая использование технологии АТМ в качестве транспорта канального уровня.

Следующий уровень (уровень III) - это уровень межсетевого взаимодействия, который занимается передачей дейтаграмм с использованием различных локальных сетей, территориальных сетей X.25, линий специальной связи и т. п. В качестве основного протокола сетевого уровня в стеке используется протокол IP, который изначально проектировался как протокол передачи пакетов в составных сетях, состоящих из большого количества локальных сетей, объединенных как локальными, так и глобальными связями.

Поэтому протокол IP хорошо работает в сетях со сложной топологией, рационально используя наличие в них подсистем и экономно расходуя пропускную способность низкоскоростных линий связи.

К уровню межсетевого взаимодействия относятся и все протоколы, связанные с составлением и модификацией таблиц маршрутизации, такие как протоколы сбора маршрутной информации RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First), а также протокол межсетевых управляющих сообщений ICMP (Internet Control Message Protocol). Последний протокол предназначен для обмена информацией об ошибках между маршрутизатором и шлюзом, системой-источником и системой-приемником, то есть для организации обратной связи. С помощью специальных пакетов ICMP сообщается о невозможности доставки пакета, о превышении времени жизни или продолжительности сборки пакета из фрагментов, об аномальных величинах параметров, об изменении маршрута пересылки и типа обслуживания, о состоянии системы и т.п.

Следующий уровень (уровень II) называется транспортным. Протоколы транспортного уровня обеспечивают прозрачную (сквозную) доставку данных (end-to-end delivery service) между двумя прикладными процессами. Процесс, получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне номером, который называется номером порта. Таким образом, роль адреса отправителя и получателя на транспортном уровне выполняет номер порта (или проще - порт).

Анализируя заголовок своего пакета, полученного от межсетевого уровня, транспортный модуль определяет по номеру порта получателя, какому из прикладных процессов направлены данные, и передает эти данные соответствующему прикладному процессу (возможно, после проверки их на наличие ошибок и т.п.). Номера портов получателя и отправителя записываются в заголовок транспортным модулем, отправляющим данные; заголовок транспортного уровня содержит также и другую служебную информацию; формат заголовка зависит от используемого транспортного протокола.

На транспортном уровне работают два основных протокола: UDP (User Datagram Protocol) и TCP (Transmission Control Protocol).

TCP (Transmission Control Protocol - протокол контроля передачи) - надежный протокол с установлением соединения: он управляет логическим сеансом связи (устанавливает, поддерживает и закрывает соединение) между процессами и обеспечивает надежную (безошибочную и гарантированную) доставку прикладных данных от процесса к процессу.

Протокол UDP обеспечивает передачу прикладных пакетов дейтаграммным методом, то есть без установления виртуального соединения, и поэтому требует меньших накладных расходов, чем TCP.

Данными для TCP является не интерпретируемая протоколом последовательность пользовательских октетов, разбиваемая для передачи по частям. Каждая часть передается в отдельном TCP-сегменте. Для продвижения сегмента по сети между компьютером-отправителем и компьютером-получателем модуль TCP пользуется сервисом межсетевого уровня (вызывает модуль IP).

Самый верхний уровень (уровень I) называется прикладным уровнем. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и сервисов прикладного уровня. Для пересылки данных другому приложению, приложение обращается к тому или иному модулю транспортного уровня.

Протокол SNMP (Simple Network Management Protocol) используется для организации сетевого управления. Проблема управления разделяется здесь на две задачи. Первая задача связана с передачей информации. Протоколы передачи управляющей информации определяют процедуру взаимодействия сервера с программой-клиентом, работающей на хосте администратора. Они определяют форматы сообщений, которыми обмениваются клиенты и серверы, а также форматы имен и адресов. Вторая задача связана с контролируемыми данными. Стандарты регламентируют, какие данные должны сохраняться и накапливаться в шлюзах, имена этих данных и синтаксис этих имен.

Протокол пересылки файлов FTP (File Transfer Protocol) реализует удаленный доступ к файлу. Для того, чтобы обеспечить надежную передачу, FTP использует в качестве транспорта протокол с установлением соединений - TCP. Кроме пересылки файлов протокол, FTP предлагает и другие услуги. Так пользователю предоставляется возможность интерактивной работы с удаленной машиной, например, он может распечатать содержимое ее каталогов, FTP позволяет пользователю указывать тип и формат запоминаемых данных. Наконец, FTP выполняет аутентификацию пользователей. Прежде, чем получить доступ к файлу, в соответствии с протоколом пользователи должны сообщить свое имя и пароль.

Приложения, которым не требуются все возможности FTP, могут использовать другой, более экономичный протокол - простейший протокол пересылки файлов TFTP (Trivial File Transfer Protocol). Этот протокол реализует только передачу файлов, причем в качестве транспорта

используется более простой, чем TCP, протокол без установления соединения - UDP.

Протокол telnet обеспечивает передачу потока байтов между процессами, а также между процессом и терминалом. Наиболее часто этот протокол используется для эмуляции терминала удаленной ЭВМ. При использовании сервиса telnet пользователь фактически управляет удаленным компьютером так же, как и локальный пользователь, поэтому такой вид доступа требует хорошей защиты.

Рассмотрев кратко все протоколы прикладного уровня стека TCP/IP, остановимся подробнее на рассмотрении протокола передачи файлов FTP (File Transfer Protocol).

### 1.3.3 Служба архивов FTP

Технология FTP (от протокола FTP - File Transfer Protocol) была разработана в рамках проекта ARPA и предназначена для обмена большими объемами информации между машинами с различной архитектурой. Стержень технологии составляет FTP-протокол.

**Служба FTP** - предназначена для обмена файлами и построена по архитектуре "клиент-сервер".

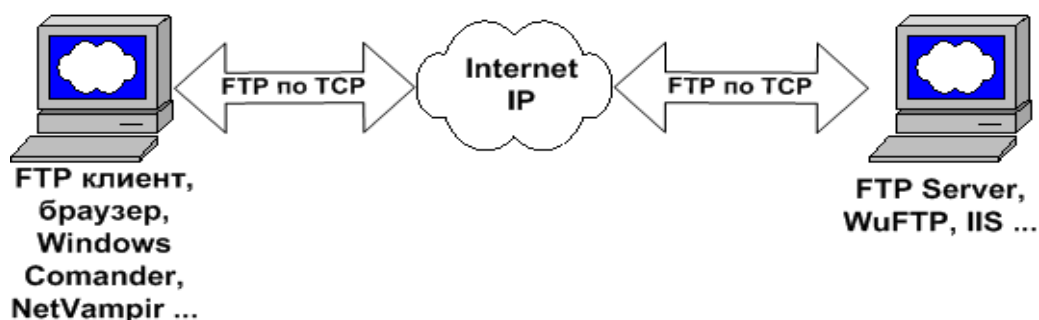


Рисунок 6 - Работа FTP на пользовательском уровне

Клиент посылает запросы серверу и принимает файлы. Сервер обрабатывает запросы клиента на получение файла.

Служба FTP базируется на двух стандартах:

- URL (Universal Resource Locator) - универсальный способ адресации ресурсов в сети;
- FTP (File Transfer Protocol) - протокол передачи файлов.

Необходимо отметить, что FTP-архивы являются одним из основных информационных ресурсов Internet. Фактически, это распределенное хранилище текстов, программ, фильмов, фотографий, аудио записей и прочей информации, хранящейся в виде файлов на различных компьютерах во всем мире.

Все ftp-серверы делятся на три большие группы:

1 Защищенная информация, режим доступа к которой определяется ее владельцами и разрешается по специальному соглашению с потребителем. К этому виду ресурсов относятся коммерческие архивы, закрытые

некоммерческие ресурсы, частная некоммерческая информация (частные благотворительные фонды).

2 Вторая группа ftp-серверов является более распространенной. Это вполне легальные архивы программ и данных, доступ на которые извне разрешен исключительно какой-либо определенной группе пользователей, которые формируются по тому или иному признаку (клиенты определенного провайдера, сотрудники компании). При входе пользователи предъявляют пропуск – то есть пароль. В данный класс могут входить ресурсы ограниченного времени использования (текущая версия Netscape перестанет работать в июне, если только кто-то не сломает защиту) или ограниченного времени действия, т.е. пользователь может использовать текущую версию, но никто не будет оказывать ему поддержку.

3 Свободно распространяемые информационные ресурсы. К этим ресурсам относится все, что можно свободно получить по сети без специальной регистрации. Как правило, это серверы содержат архивы разнообразных бесплатных и условно-бесплатных программ. Однако свободно распространяемое программное обеспечение не имеет сертификата качества, но, как правило, его разработчики открыты для обмена опытом.

Из выше перечисленных ресурсов наиболее интересными, по понятным причинам, являются две последних категории, которые, как правило, оформлены в виде FTP-архивов.

Когда не очень опытный пользователь попадает на ftp-сервер, он теряется при виде огромного количества директорий. Необходимо обратиться к директории pub.

Многие пользователи пугаются момента, когда при клике мышью по папке pub/win98 появляется список из 400 поддиректорий, в каждой из которых по тысяче файлов. На всех публичных ftp-серверах в каждой директории лежит файл (что-нибудь вроде list.txt или info.zip), где подробно расписано все содержимое. Его надо скачать и просмотреть.

Некоторые предпочитают производить поиск по архивам исключительно вручную. Как правило, подобные ситуации возникают при поиске либо чего-то специфического, что лежит на сервере, не прописанном ни в одной поисковой системе.

### **1.3.3.1 Протокол FTP**

Протокол FTP, как и любой протокол прикладного уровня использует в качестве транспорта протокол TCP/IP. Протокол TCP разбивает информацию на порции и нумерует все порции, чтобы при получении можно было правильно собрать информацию. Далее с помощью протокола IP все части передаются получателю, где с помощью протокола TCP проверяется, все ли части получены. Так как отдельные части могут путешествовать по Интернет самыми разными путями, то порядок прихода частей может быть нарушен. После получения частей TCP располагает их в нужном порядке и собирает в единое целое.

Для протокола TCP не имеет значения, какими путями информация путешествует по Интернет. Этим занимается протокол IP. К каждой полученной порции информации протокол IP добавляет служебную информацию, из которой можно узнать адреса отправителя и получателя информации. Если следовать аналогии с почтой, то данные помещаются в конверт, на котором пишется адрес получателя.

Далее протокол IP, так же как и обычная почта, обеспечивает доставку всех конвертов получателю.

При этом скорость и пути прохождения разных конвертов могут быть различными. Если при путешествии отдельного конверта наблюдались помехи, и информация пришла искаженной, следует повторный запрос об отправке искаженной части до тех пор, пока она не будет принята без искажений (в этом еще один плюс приема–передачи информации порциями).

Протокол FTP появился тогда, когда еще сеть называлась ARPANET. Он предназначался исключительно для передачи файлов с компьютера А на компьютер Б, а главным его достоинством было и по сей день является то, что этому протоколу совершенно все равно, с какой и на какую платформу передавать файл, а также безразлично, какая сеть для передачи используется.

Интересен протокол FTP тем, что, фактически, он использует одновременно сразу 2 TCP-соединения для передачи одного файла: первое употребляется для управления процессом, а второе - непосредственно для передачи данных:

1) Управляющее соединение - соединение для посылки команд серверу и получение ответов от него. Для канала управления используется протокол Telnet.

2) Соединение данных - соединение для передачи файлов (рисунок 7).

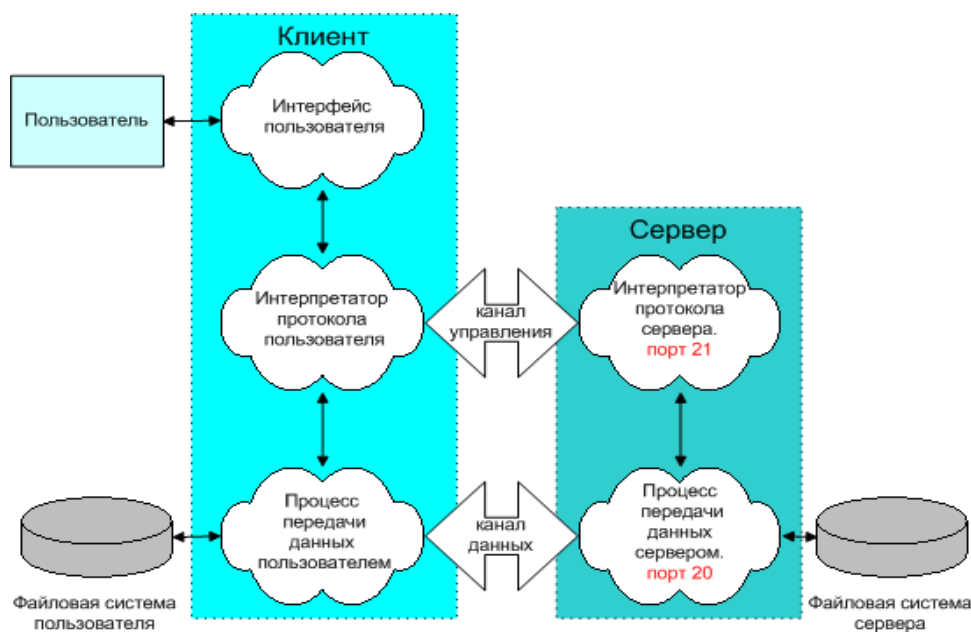


Рисунок 7 – Работа протокола FTP

FTP-соединение инициируется интерпретатором протокола пользователя. Управление обменом осуществляется по каналу управления в

стандарте протокола Telnet. Команды FTP генерируются интерпретатором протокола пользователя и передаются на сервер. Ответы сервера отправляются пользователю также по каналу управления. В общем случае пользователь имеет возможность установить контакт с интерпретатором протокола сервера и отличными от интерпретатора пользователя средствами.

Команды FTP определяют параметры канала передачи данных и самого процесса передачи. Они также определяют и характер работы с удаленной и локальной файловыми системами. При организации канала передачи данных последовательность действий другая, отличная от организации канала управления. В этом случае сервер инициирует обмен данными в соответствии с согласованными в сессии управления параметрами.

Канал данных устанавливается для того же хоста, что и канал управления, через который ведется настройка канала данных. Канал данных может быть использован как для приема, так и для передачи данных.

Возможна ситуация, когда данные могут передаваться на третью машину. В этом случае пользователь организует канал управления с двумя серверами и организует прямой канал данных между ними. Команды управления идут через пользователя, а данные напрямую между серверами.

Канал управления должен быть открыт при передаче данных между машинами. В случае его закрытия передача данных прекращается.

Протокол FTP определяет запрос-ответный способ взаимодействия между программой-клиентом и программой-сервером.

Работа FTP на пользовательском уровне содержит несколько этапов:

1. Идентификация (ввод имени и пароля).
2. Выбор каталога.
3. Определение режима обмена (поблочный, поточный, ascii или двоичный).
4. Выполнение команд обмена (get, mget, dir, mdel, mput или put).
5. Завершение процедуры (quit или close).

В старых версиях для передачи данных использовался только 20-й порт (активный режим), в современных версиях FTP-серверов порт для канала данных может назначаться сервером из нестандартных ( $N > 1024$ ) портов (пассивный режим).

### **Различие пассивного и активного режима работы протокола FTP.**

#### **Активный режим.**

Действия сервера и клиента:

Клиент устанавливает связь и посылает запрос на 21 порт сервера с порта  $N$  ( $N > 1024$ );

Сервер посылает ответ на порт  $N$  ( $N > 1024$ ) клиента;

Сервер устанавливает связь для передачи данных по порту 20 на порт клиента  $N+1$  (рисунок 8).

#### **Пассивный режим. Действия сервера и клиента:**

1) Клиент устанавливает связь и посылает запрос (сообщает, что надо работать в пассивном режиме) на 21 порт сервера с порта  $N$  ( $N > 1024$ );

2) Сервер посылает ответ и сообщает номер порта для канала данных  $P$

( $P > 1024$ ) на порт  $N$  ( $N > 1024$ ) клиента;

3) Клиент устанавливает связь для передачи данных по порту  $N+1$  на порт сервера  $P$  ( $P > 1024$ ) (рисунок 9).

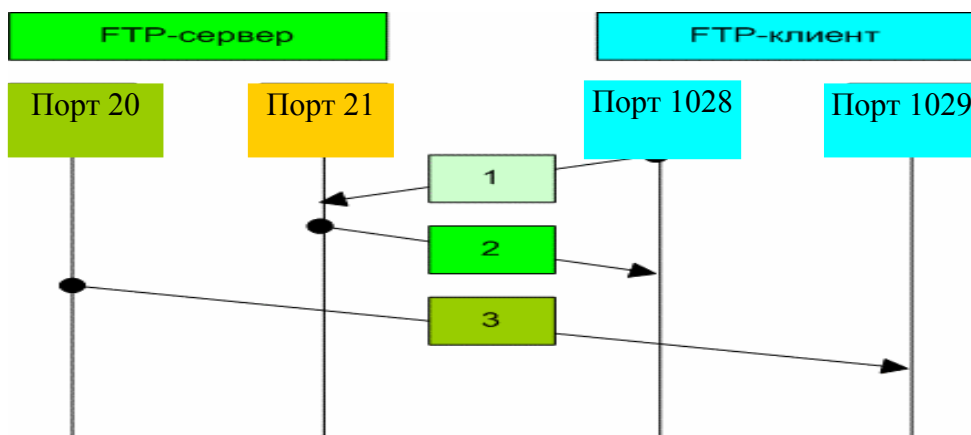


Рисунок 8 - Активный режим работы FTP

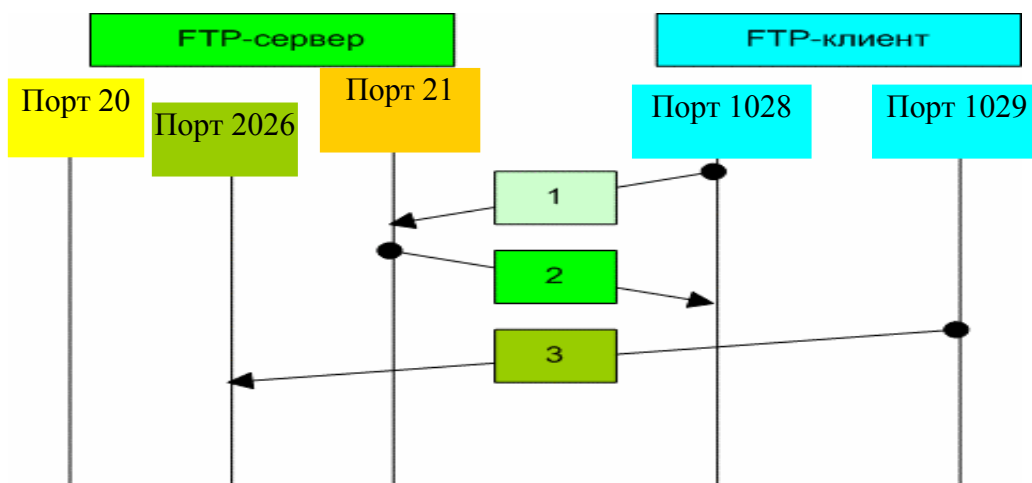


Рисунок 9 - Пассивный режим работы FTP

Активный FTP выгоден для FTP-сервера, но вреден для стороны клиента. FTP-сервер пытается соединиться со случайными высокими (по номеру) портами на клиенте, такое соединение, наверняка, будет заблокировано на стороне клиента.

Пассивный FTP выгоден для клиента, но вреден для FTP-сервера. Клиент будет делать оба соединения к серверу, но одно из них будет к случайному высокому порту, такое соединение, наверняка, будет заблокировано на стороне сервера.

#### 1.4 Контрольные вопросы

- 1) Принцип работы технологии “клиент-сервер”.
- 2) Ранжирование клиентов по “толщине”.
- 3) Понятие “стек протоколов”. Популярные стеки протоколов.
- 4) Стек TCP/IP. Основные уровни.
- 5) Соответствие уровней стека TCP/IP модели ISO/OSI.

- 6) Особенности межсетевого и транспортного уровней стека TCP/IP.
- 7) Прикладной уровень стека TCP/IP. Протоколы FTP, telnet, SMTP, TFTP.
- 8) Служба FTP. FTP-архивы.
- 9) Протокол FTP. Использование двух TCP-соединений для передачи данных.
- 10) Различие работы пассивного и активного режима протокола FTP.

### 1.5 Тесты

- 1) Принцип архитектуры “клиент-сервер”:
  - а) существует выделенный сервер, предоставляющий всевозможные сервисы, и множество клиентских ПК, использующих их в своих целях;
  - б) каждый ПК является как сервером, так и клиентом;
  - в) ни один из ПК не обладает полномочиями сервера.
  
- 2) Что такое редиректор ОС?
  - а) клиентский модуль;
  - б) серверный модуль;
  - в) модуль, который распознает запросы к удаленным ресурсам;
  - г) модуль, отвечающий за администрирование пользователей.
  
- 3) Как называется совокупный сервис, предоставляемый данным уровнем выше лежащему уровню?
  - а) стек;
  - б) протокол;
  - в) интерфейс;
  - г) стеком протоколов;
  - д) нет правильного ответа.
  
- 4) Из скольких уровней состоит стек TCP/IP?
  - а) 7;
  - б) 6;
  - в) 5;
  - г) 4
  
- 5) Какой уровень стека TCP/IP соответствует физическому и канальному уровням модели OSI?
  - а) уровень доступа к среде передачи;
  - б) межсетевой;
  - в) транспортный;
  - г) прикладной.
  
- 6) Какие протоколы относятся к транспортному уровню стека TCP/IP?
  - а) TCP;



- б) IP;
- в) FTP;
- г) UDP;
- д) ICMP;
- е) RIP.

7) На каком уровне стека протокола TCP/IP находится протокол IP?

- а) прикладной;
- б) сеансовый;
- в) транспортный;
- г) межсетевой.

8) Основная задача, решаемая протоколом IP:

- а) маршрутизация;
- б) добавление заголовка;
- в) анализ правильности доставки.

9) К какому уровню стека TCP/IP относится протокол FTP?

- а) прикладной;
- б) физический;
- в) сеансовый;
- г) транспортный.

10) Какой порт (порты) использует протокол ftp?

- а) 20;
- б) 80;
- в) 21;
- г) 23.

## Список использованных источников

- 1 Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. - СПб.: Издательство "Питер", 2000. - 672 с.
- 2 Бройдо В.Л. Вычислительные системы, сети и телекоммуникации. – СПб.: Питер, 2003. – 688.
- 3 Основы сетей передачи данных /В.Г. Олифер, Н.А. Олифер. – М.: ИНТУИТ.РУ “Интернет-Университет Информационных технологий”, 2003 – 248 с.
- 4 Пятибратов А.П. и др. Вычислительные системы, сети и телекоммуникации: Учебник/ Под редакцией А.П. Пятибратова. – М.: Финансы и статистика, 2001. – 512 с.
- 5 Олифер В.Г., Олифер Н.А. Базовые технологии локальных сетей. <http://www.citforum.ru>.
- 6 Комер Д. Межсетевой обмен с помощью TCP/IP. <http://lemoi-www.dvgu.ru/lect/protoc/tcpip/comer/contents.htm>.
- 7 Брежнев А.Ф., Смелянский Р.Л. Семейство протоколов TCP/IP. <http://lemoi-www.dvgu.ru/lect/protoc/tcpip/tcpip/index.htm>.
- 8 Усманов Р. Спецификация протокола TCP. <http://lemoi-www.dvgu.ru/lect/protoc/tcpip/tcpspec/tcpspec.htm>.
- 9 Блэк Ю. Сети ЭВМ: протоколы, стандарты, интерфейсы.- М.: Мир, 1990.
- 10 Олифер В.Г., Олифер Н.А. Новые технологии и оборудование IP-сетей. - СПб, 2000. - 512 с.

## **Приложение А**

*(справочное)*

### Задание на лабораторную работу

Написать программу ftp-клиент, которая осуществляет подключение к ftp-серверу, закидывает файлы на ftp-сервер, просматривает файлы, каталоги, удаляет файлы, каталоги.

Для тестирования программы использовать Localhost 127.0.0.1.

## Приложение Б

(справочное)

### Ответы на тесты

Номер вопроса	Ответ
1	а
2	а
3	в
4	Г
5	а
6	а, Г
7	Г
8	а
9	а
10	а,в

## Приложение В

(справочное)

### Правила присвоения классификационного кода

	X	XXXXXX.	X	X	XX.	XX	XXX
Код организации-разработчика (ГОУ ОГУ)							
Шифр специальности (190600, 060400 и т.д.)							
Код вида документации							
Дипломный проект –1							
Дипломная работа –2							
Дипломная работа для нетехнических специальностей –3							
Курсовой проект – 4							
Курсовая работа –5							
РГР – 6							
УИРС –7							
Реферат – 8							
Практика –9							
Характеристика тем							
Без указания –0							
Конструкторская –1							
Технологическая –2							
Исследовательская – 3							
Комбинированная – 4							
Год издания работы							
Обозначается двумя последними цифрами календарного года, в котором защищается проект (работа, реферат)							
Порядковый номер исполнителя.							
Берется по журналу данной группы, в котором список студентов приведен в алфавитном порядке							
Шифр документа							
ПЗ –пояснительная записка							
О – отчет по РГР							
У – отчет по УИРС							
Р – реферат							
П – отчет по практике							
ОО – для нетехнических специальностей							

# Приложение Г

(справочное)

## Пример оформления содержания

Содержание	
Введение.....	3
1 Общие сведения о программном средстве.....	5
1.1 Основное функциональное назначение программного средства.....	6
1.2 Полное наименование программного средства.....	6
1.3 Условное обозначение программного средства.....	7
1.4 Разработчики программного средства.....	7
2 Техническое задание.....	8
2.1 Основание для разработки.....	9
2.2 Назначение разработки.....	9
2.3 Требования к функциям, выполняемым программным средством.....	10
2.4 Требования к программному средству.....	11
2.5 Требования к программной документации.....	12
2.6 Требования к эргономике и технической эстетике.....	12
2.7 Стадии и этапы разработки.....	13
2.8 Порядок контроля и приемки.....	14
3 Пояснительная записка.....	15
и т.д.	
Заключение.....	67
Список использованных источников.....	68
Приложение А Контрольные примеры.....	69
Приложение Б Текст программы.....	75

	Лист
	4

**Приложение Д**  
*(обязательное)*  
**Структура отчета**

- 1 Титульный лист.
- 2 Постановка задачи.
- 3 Содержание.
- 4 Теоретические предпосылки.
- 5 Список использованных источников.
- 6 Программный код.
- 7 Схема алгоритма программы и ее основных модулей.
- 8 Контрольный пример работы программы.

## **Приложение Е**

*(справочное)*

### **Пример оформления отчета по лабораторной работе**

Министерство образования и науки Российской Федерации

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ОБРАЗОВАНИЯ**

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
“ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет информационных технологий

Кафедра программного обеспечения вычислительной техники  
и автоматизированных систем

### **Лабораторная работа**

по дисциплине “Сети ЭВМ и телекоммуникации”

“FTP-клиент”

ГОУ ОГУ 220400.9004.18 О

Руководитель работы

\_\_\_\_\_ Гайсина Л.Ф.

" \_\_\_\_ " \_\_\_\_\_ 2005г.

Исполнитель

студент гр. 00ПО1

\_\_\_\_\_ Иванов Д.И.

" \_\_\_\_ " \_\_\_\_\_ 2005г.

Оренбург 2005



## **Постановка задачи**

Написать программу ftp-клиент, которая осуществляет подключение к ftp-серверу, закичивает файлы на ftp-сервер, просматривает файлы, каталоги, удаляет файлы, каталоги.

## Содержание

1 Теоретические предпосылки .....	4
Список использованных источников .....	6
Приложение А Программный код .....	7
Приложение Б Схема алгоритма.....	8
Приложение В Контрольный пример.....	10

## 1 Теоретические предпосылки

FTP (File Transfer Protocol или "Протокол Передачи Файлов") - один из старейших протоколов в Internet и входит в его стандарты. Обмен данными в FTP проходит по TCP-каналу. Построен обмен по технологии "клиент-сервер". На рисунке 1 изображена модель протокола.

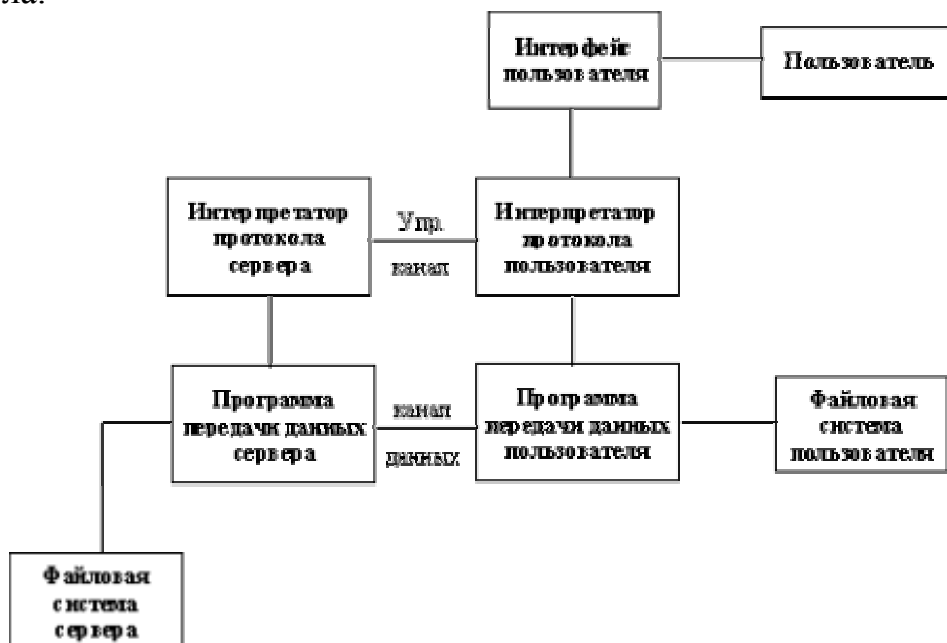


Рисунок 1 – Модель протокола ftp

В FTP соединение инициируется интерпретатором протокола пользователя. Управление обменом осуществляется по каналу управления в стандарте протокола TELNET. Команды FTP генерируются интерпретатором протокола пользователя и передаются на сервер. Ответы сервера отправляются пользователю также по каналу управления. В общем случае пользователь имеет возможность установить контакт с интерпретатором протокола сервера и отличными от интерпретатора пользователя средствами. Команды FTP определяют параметры канала передачи данных и самого процесса передачи. Они также определяют и характер работы с удаленной и локальной файловыми системами. Сессия управления инициализирует канал передачи данных. При организации канала передачи данных последовательность действий другая, отличная от организации канала управления. В этом случае сервер инициирует обмен данными в соответствии с параметрами, согласованными в сессии управления. Канал данных устанавливается для того же host'a, что и канал управления, через который ведется настройка канала данных. Канал данных может быть использован как для приема, так и для передачи данных. Возможна ситуация, когда данные

В этом случае пользователь организует канал управления с двумя серверами и организует прямой канал данных между ними. Команды управления идут через пользователя, а данные напрямую между серверами (рисунок 2). Канал управления должен быть открыт при передаче данных между машинами. В случае его закрытия передача данных прекращается.



Рисунок 2 – Соединение с двумя разными серверами и передача данных между ними

В протоколе большое внимание уделяется различным способам обмена данными между машинами различных архитектур. Действительно, чего только нет в Internet, от персоналок и Mac'ов до суперкомпьютеров. Все они имеют различную длину слова и многие различный порядок битов в слове. Кроме этого, различные файловые системы работают с разной организацией данных, которая выражается в понятии метода доступа. В общем случае, с точки зрения FTP, обмен может быть поточный или блочный, с кодировкой в промежуточные форматы или без нее, текстовый или двоичный. При текстовом обмене все данные преобразуются в ASCII и в этом виде передаются по сети. Исключение составляют только данные IBM mainframe, которые по умолчанию передаются в EBCDIC, если обе взаимодействующие машины IBM.

## Список использованных источников

- 1 Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. - СПб.: Издательство "Питер", 2000. - 672 с.
- 2 Бройдо В.Л. Вычислительные системы, сети и телекоммуникации. – СПб.: Питер, 2003. – 688.
- 3 Пятибратов А.П. и др. Вычислительные системы, сети и телекоммуникации: Учебник/ Под редакцией А.П. Пятибратова. – М.: Финансы и статистика, 2001. – 512 с.
- 4 Комер Д. Межсетевой обмен с помощью TCP/IP. <http://lemoi-www.dvgu.ru/lect/protoc/tcpip/comer/contents.htm>.
- 5 Брежнев А.Ф., Смелянский Р.Л. Семейство протоколов TCP/IP. <http://lemoi-www.dvgu.ru/lect/protoc/tcpip/tcpip/index.htm>.
- 6 Усманов Р. Спецификация протокола TCP. <http://lemoi-www.dvgu.ru/lect/protoc/tcpip/tcpspec/tcpspec.htm>.
- 7 Блэк Ю. Сети ЭВМ: протоколы, стандарты, интерфейсы.- М.: Мир, 1990.
- 8 Олифер В.Г., Олифер Н.А. Новые технологии и оборудование IP-сетей. - СПб, 2000. - 512 с.

# Приложение А

## (справочное)

### Программный код

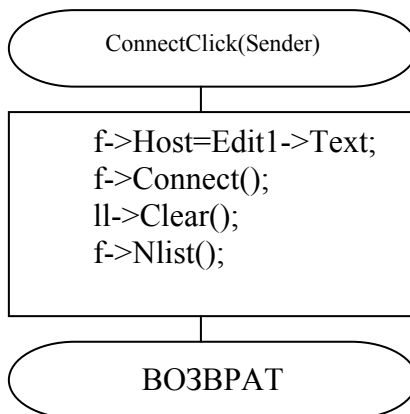
```

#include <vcl.h>
#pragma hdrstop
#include "Uftp.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int i,k=0;
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
void __fastcall TForm1::ConnectClick(TObject *Sender)
{
try
{
f->Host=Edit1->Text;
f->Connect();
ll->Clear();
f->Nlist();
}
catch(ESockError&)
{
ShowMessage("Άίçîîæíí ñãêþ÷àíèà ïðìèçìøëï"
" èèè àù íà ñãêþ÷àíù è èíèèëüíé ñàðè ");
}
}
void __fastcall TForm1::DisconnectClick(TObject
*Sender)
{
f->Disconnect();
}
void __fastcall TForm1::fListItem(AnsiString Listing)
{
ll->Items->Add(Listing);
}
void __fastcall TForm1::DownFClick(TObject *Sender)
{
if (f->Connected)
if (SaveDialog1->Execute())
{
if (ll->Items->Strings[ll->ItemIndex]!="")
{SaveDialog1->FileName=ll->Items->Strings[ll-
>ItemIndex];
f->Download(ll->Items->Strings[ll-
>ItemIndex],SaveDialog1->FileName);
}
}
else ShowMessage("Ïãêþ÷àíèà íà ïðìèçìøëï");
}
void __fastcall TForm1::UploadClick(TObject *Sender)
{
String s,d,ff;
if (f->Connected)
{
if (oo->Execute()
{
d=oo->FileName;
for (i=d.Length(); i>=1 ; i--)
{if (d[i]!='\')
{s+=d[i];} else {break;}
}
ff=s;
for (i=s.Length(); i>=1 ; i--)
{ff[s.Length()-i+1]=s[i];}
Edit3->Text=ff;
f->Upload(oo->FileName,ff);
ll->Items->Add(ff);
}
} else ShowMessage("Ïãêþ÷àíèà íà ïðìèçìøëï");
}
void __fastcall TForm1::DeleteFClick(TObject *Sender)
{
if (f->Connected)
{
if (ll->Items->Strings[ll->ItemIndex]!="")
{
f->Delete(ll->Items->Strings[ll->ItemIndex]);
ll->Items->Delete(ll->ItemIndex);
}
} else ShowMessage("Ïãêþ÷àíèà íà ïðìèçìøëï");
}
void __fastcall TForm1::llDbClick(TObject *Sender)
{
try
{
if (ll->Items->Strings[ll->ItemIndex]==".."){i=1;}
if (i!=0)
{
if (ll->Items->Strings[ll->ItemIndex]=="..") {k--;}
else {k++;}
f->ChangeDir(ll->Items->Strings[ll->ItemIndex]);
ll->Clear();
if (k!=0)
{ll->Items->Add("..");}
f->Nlist();
}
}
catch(...)
{
ShowMessage("Άίçîîæíí Âù ïùðààðàñù 'çàéòè"
" à ðàèè, ÿòíàí íáëüçÿ, äëÿ ñèà÷èàíéÿ íàæìèðà èííèð
"Ñèà÷èà");
}
}
void __fastcall TForm1::FormKeyPress(TObject
*Sender, char &Key)
{
if (Key==13)
{
llDbClick(Sender);
}
}

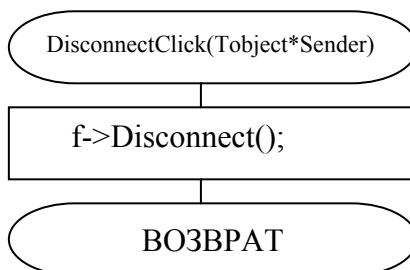
```

## Приложение Б (справочное) Схема алгоритма

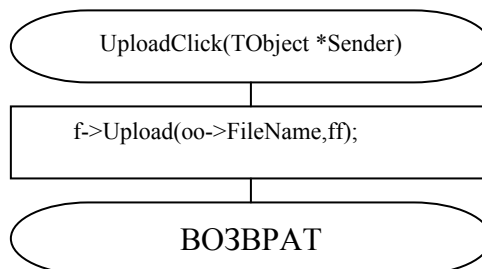
Функция подключения к FTP – серверу и вывод корневого каталога на экран



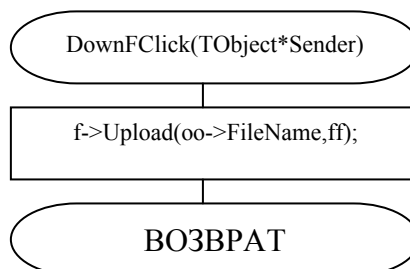
Функция отключения от FTP – сервера.



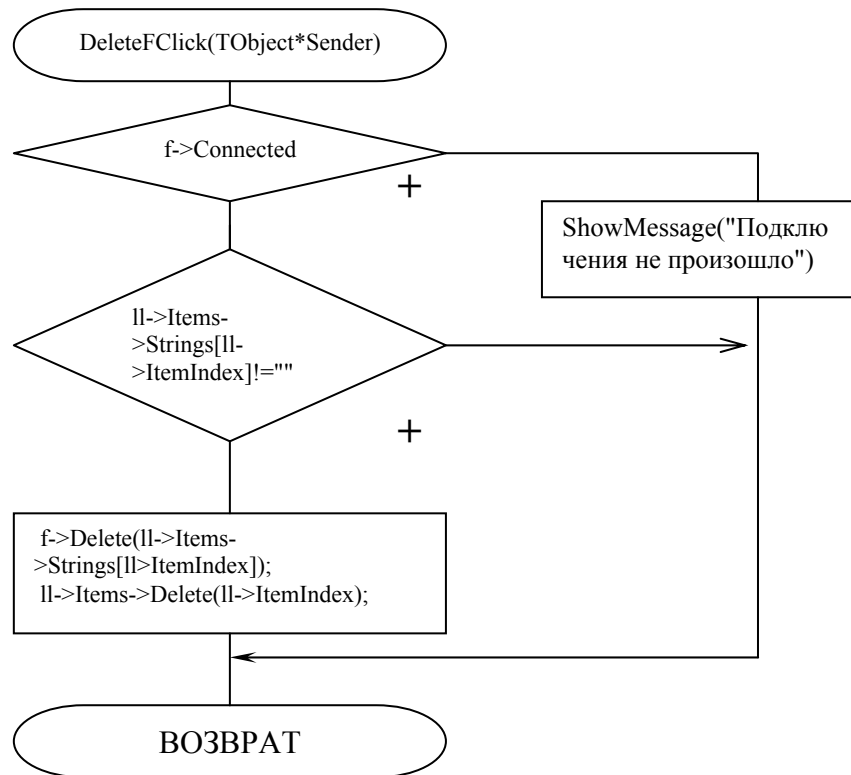
Функция загрузки файла на FTP – сервер



Функция скачивания файла с FTP – сервера



## Функция удаления файла с FTP – сервера





## Приложение В (справочное) Контрольный пример

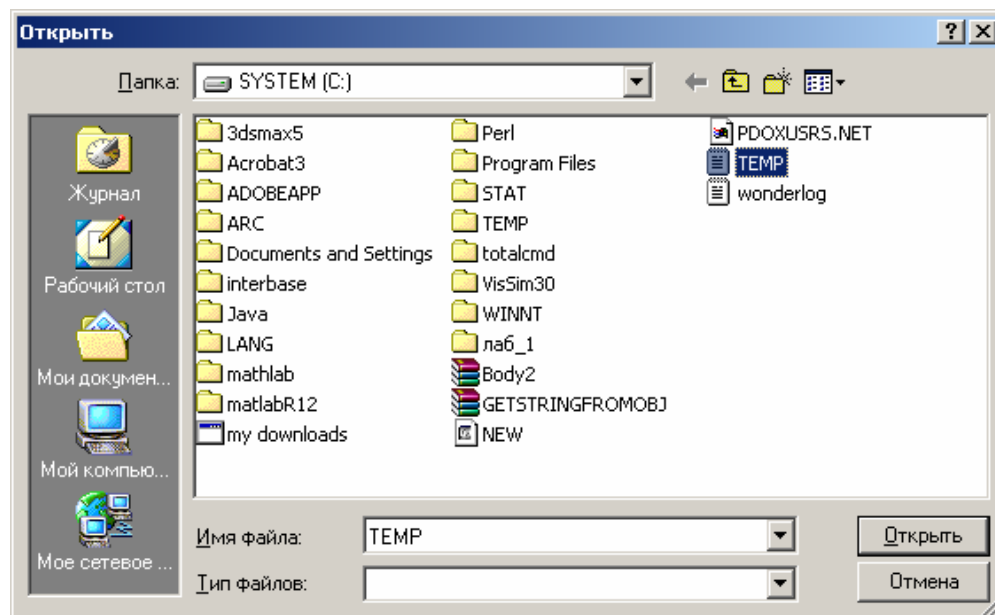


Рисунок В.1– Окно выбора для копирования файла на сервер

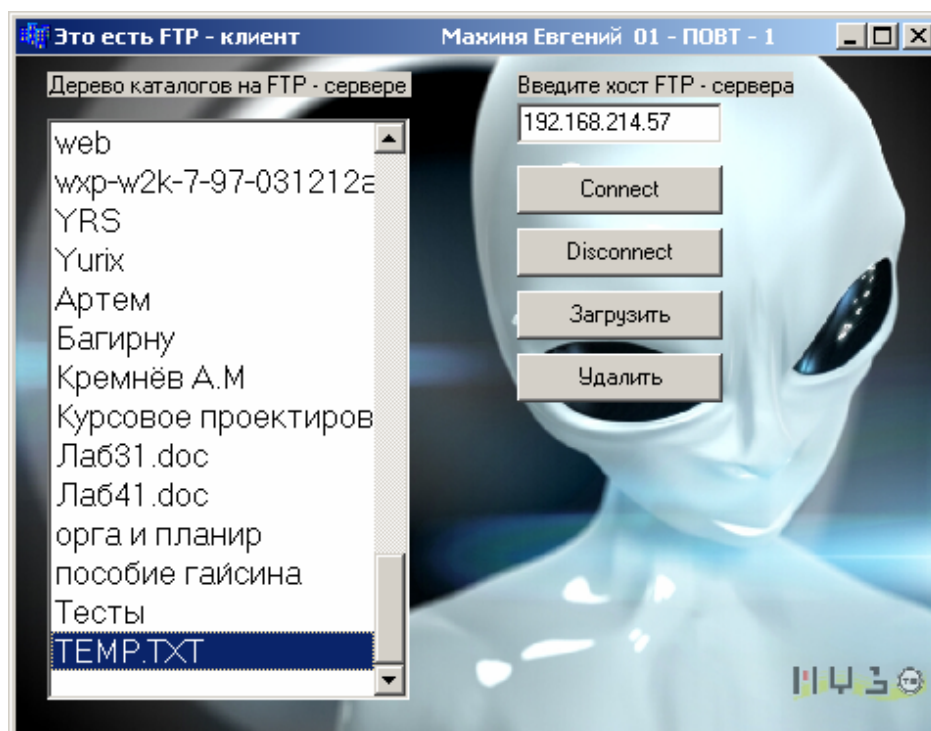


Рисунок В.2– Скопированный файл на FTP – сервере

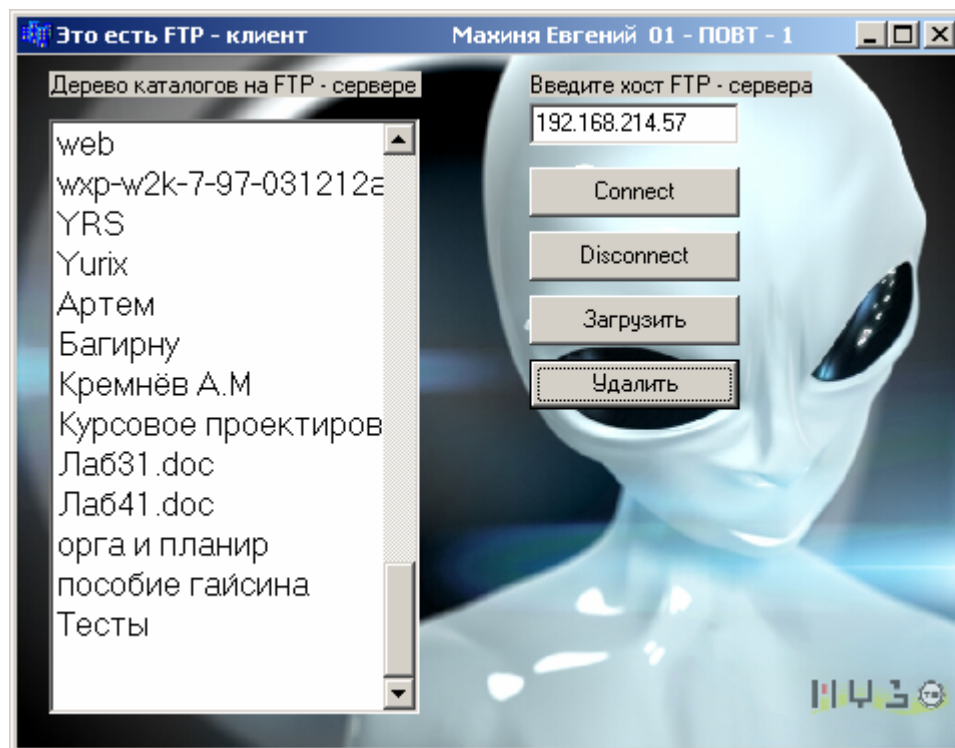


Рисунок В.3 – Дерево папок FTP – сервера после удаления файла