Министерство науки и высшего образования Российской Федерации

Калужский филиал федерально-
го государственного бюджетного образовательного учреждения высшего обр
азования
«Московский государственный технический университет имени Н.Э. Баума-
на (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

Е.В.Вершинин, М.Е. Смирнов

РАЗРАБОТКА СЕТЕВЫХ ПРИЛОЖЕНИЙ

Методические указания по выполнению
домашней работы по дисциплине "Сети и телекоммуникации"

Калуга, 2020

УДК 004.7
ББК 32.973.202

Данные методические указания издаются в соответствии с учебным планом кафедры «Системы обработки информации» для направления подготовки 09.03.01 «Информатика и вычислительная техника» КФ МГТУ им. Н.Э. Баумана.

Указания рассмотрены и одобрены:

Кафедрой «Системы обработки информации» (ИУК5) _____ протокол №_____

Зав кафедрой ИУК5 _____ Е.В.Вершинин

Методической комиссией ИУК _____ протокол №_____

Председатель методической комиссии ИУК _____ М.Ю.Адкин

Методической комиссией Калужского филиала _____ протокол № ____

Председатель методической комиссии_____ О.Л. Перерва

Рецензент:
к. т. н., зав. кафедрой «Информационные системы и сети» (ИУК2-КФ)
_____ И.В.Чухраев

Авторы:
к. ф.-м. н., доцент кафедры ИУК5 _____ Е.В. Вершинин

ст. преподаватель ИУК5 _____ М.Е. Смирнов

Аннотация

Методические указания по выполнению домашней работы "Разработка сетевых приложений" по курсу «Сети и телекоммуникации» содержат требования и рекомендации по выполнению, оформлению и защите работы.

Предназначены для студентов 3-го курса КФ МГТУ им Н.Э. Баумана, обучающихся по направлению подготовки 09.03.01 «Информатика и вычислительная техника».

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Домашняя работа представляет собой разработку клиент-серверного приложения для ОС Windows, обеспечивающего требуемую функциональность. Выполненная работа способствует углубленному изучению основных, наиболее трудных и важных разделов дисциплины "Сети и телекоммуникации". Домашняя работа является важным этапом в подготовке к выполнению курсовой работы по дисциплине Сетевые технологии в АСОИУ.

Данные указания предоставляют студенту возможность правильно и квалифицированно выполнить домашнюю работу, соблюдая при этом все требования по её оформлению. Творческое развитие логики, использование возможностей современных операционных систем и средств их администрирования, выполнение и защита в установленные сроки поставленных задач — всё это является залогом соответствия предъявляемым ФГОС требованиям к подготовке бакалавров.

Указания предназначены для студентов 3-го курса бакалавриата КФ МГТУ им Н.Э. Баумана, обучающихся по направлению подготовки 09.03.01«Информатика и вычислительная техника».

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью домашней работы является получение практических навыков по разработке клиент-серверных приложений, использующих для связи механизм сокетов.

Задачи:

1. Создать приложение-сервер, предназначенное для параллельной обработки запросов и работающее в ОС Windows или *NIX.
2. Создать приложение-клиент, которое будет подключаться к серверу с удаленных компьютеров. Рекомендуется использовать интерфейс сокетов и протокол TCP.
3. Разработать и реализовать протокол прикладного уровня для взаимодействия приложений.
4. Обеспечить обмен произвольными данными между клиентом и сервером.

Результатами работы являются:

- клиент-серверное приложение, обеспечивающее заданный функционал;
- подготовленный и оформленный согласно требованиям отчет о выполнении домашней работы;

КРАТКИЕТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Сокеты Беркли —интерфейс программирования приложений(API), представляющий собой библиотеку для разработки приложений на языке С с поддержкой межпроцессного взаимодействия(IPC), часто применяемый в компьютерных сетях.

Сокеты Беркли (также известные как API сокетов BSD) впервые появились как API в операционной системе 4.1BSDUnix (выпущенной в 1982 году). Тем не менее, только в 1989 году Калифорнийский университет в Беркли смог начать выпускать версии операционной системы и сетевой библиотеки без лицензионных ограничений AT&T, действующих в защищённой авторским правом Unix.

API сокетов Беркли сформировал фактически стандарт абстракции для сетевых сокетов. Большинство прочих языков программирования используют интерфейс, схожий с API языка Си.

API Интерфейса транспортного уровня (TLI), основанный на STREAMS, представляет собой альтернативу сокетному API. Тем не менее, API сокетов Беркли значительно преобладает в популярности и количестве реализаций.

Windows Sockets API (WSA), название которого было укорочено до **Winsock**. Это техническая спецификация, которая определяет, как сетевое программное обеспечение Windows будет получать доступ к сетевым сервисам, в том числе TCP/IP. Он определяет стандартный интерфейс между клиентским приложением (таким как FTP-клиент или веб-браузер) и внешним стеком протоколов TCP/IP. Он основывается на API модели сокетов Беркли, использующейся в BSD для установки соединения между программами. Все функции Winsock начинаются с префекса WSA. Но в Windows можно использовать сокеты Беркли.

Интерфейс сокета Беркли —API, позволяющий реализовывать взаимодействие между компьютерами или между процессами на одном компьютере. Данная технология может работать со множеством различных устройств ввода-вывода и драйверов, несмотря на то, что их поддержка зависит от реализации операционной системы. Подобная реализация интерфейса лежит в основе TCP/IP, благодаря чему считается одной из фундаментальных технологий, на которых основывается Интернет. Все современные операционные системы имеют ту или иную реализацию интерфейса сокетов Беркли, так как это стало стандартным интерфейсом для подключения через сеть Интернет.

Программисты могут получать доступ к интерфейсу сокетов на трёх различных уровнях, наиболее мощным и фундаментальным из

которых является уровень сырых сокетов. Довольно небольшое число приложений нуждается в ограничении контроля над исходящими соединениями, реализуемыми ими, поэтому поддержка сырых сокетов задумывалась быть доступной только на компьютерах, применяемых для разработки на основе технологий, связанных с Интернет. Впоследствии в большинстве операционных систем была реализована их поддержка.

Блокирующий и неблокирующий режимы работы

Сокеты могут работать в одном из двух режимов: блокирующем или неблокирующем. *Блокирующий* сокет не возвращает контроль, пока не отошлёт (или пока не получит) все данные, указанные для операции. Это верно лишь для Linux-систем. В других системах, например во FreeBSD, вполне естественно для блокирующего сокета посылать не все данные. Приложение должно проверять возвращаемое значение для отслеживания того, сколько байт было послано/получено и, соответственно, перепосылать необработанную на данный момент информацию. Это может привести к проблемам, если сокет продолжает «слушать»: программа может повиснуть из-за того, что сокет ждет данных, которые могут никогда не прибыть.

Сокет обычно указывается блокирующим или неблокирующим при помощи функций `fcntl()` или `ioctl()`.

При неблокирующем режиме, функция вернётся сразу, и поток не будет “висеть”. Т.к. функции работают асинхронно, необходимо опрашивать сокет (модель select или WSAAsyncSelect в Winsock), и, если в нём есть данные, вызвать функцию, которая сразу вернёт результат.

Если используется блокирующий режим, то необходимо такие вызовы помещать в отдельный поток. Но в таком случае возникает необходимость в синхронизации потоков. В Windows можно также использовать другие методы (WSAEventSelect, порты завершения).

Замечания о сетевом порядке байтов

Как правило, в современных компьютерах минимальный элемент оперативной памяти, имеющий уникальный адрес, имеет длину 8 бит (1 байт). И, кроме того, процессоры умеют манипулировать как целым несколькими байтами: двумя, четырьмя, восемью, в зависимости от разрядности процессора. Хранение в памяти двух-, четырех- и восьмибайтовых слов, рассматриваемых как знаковые или беззнаковые целые числа, можно организовать по-разному. Именно, можно хранить самый младший (наименее значимый) байт числа по меньшему адресу, а можно наоборот, по меньшему адресу хранить самый

старший (наиболее значимый) байт. Например, в процессорах семейства Intel используется первый способ, а в процессорах Motorola - второй. Поэтому, для того, чтобы компьютеры с разными в этом смысле процессорами могли обмениваться данными по сети, нужно договориться о том, в каком порядке байты будут передаваться по сети. Например, в семействе протоколов TCP/IP принят порядок, обратный по сравнению с тем, какой используется в процессорах Intel, то есть 2-х и 4-х байтовые числа должны передаваться, начиная с самого старшего байта. В этих протоколах в сетевом порядке байтов хранятся, в частности, IP-адрес и номер TCP-порта. Забота о преобразовании данных от локального порядка байтов к сетевому при передаче в сеть и от сетевого к локальному при приеме из сети лежит на программном обеспечении TCP/IP и на прикладном программисте. Как правило, среди функций, входящих в состав интерфейса прикладных программ, имеются функции для преобразования чисел из локального порядка байтов к сетевому и наоборот.

Сервер

Создание простейшего TCP-сервера состоит из следующих шагов:

- Создание TCP-сокетов вызовом функции socket().
- Привязывание сокета к прослушиваемому порту вызовом функции bind(). Перед вызовом bind() программист должен объявить структуру sockaddr_in, очистить её (при помощи memset()), затем sin_family(PF_INET или PF_INET6) и заполнить поля sin_port (прослушиваемый порт, указать в виде последовательности байтов). Преобразование shortint в порядок байтов может быть выполнено при помощи вызова функции htons() (сокращениеот«от хоста в сеть»).
- Подготовка сокета к прослушиванию на предмет соединений (создание прослушиваемого сокета) при помощи вызова `listen()`.
- Принятие входящих соединений через вызов `accept()`. Это блокирует сокет до получения входящего соединения, после чего возвращает дескриптор сокета для принятого соединения. Первоначальный дескриптор остаётся прослушиваемым дескриптором, а `accept()` может быть вызван вновь для этого сокета в любое время (пока он открыт).
- Соединение с удаленным хостом, которое может быть создано при помощи send() и recv() или write() и read().

- Итоговое закрытие каждого открытого сокета, который больше не нужен, происходит при помощи close(). Необходимо отметить, что если были любые вызовы fork(), то каждый процесс должен закрыть известные ему сокеты (ядро отслеживает количество процессов, имеющих открытый дескриптор), а кроме того, два процесса не должны использовать один и тот же сокет в одно время.

Клиент

СозданиеTCP-клиента происходит следующим образом:

- Создание TCP-сокета вызовом socket().
- Соединение с сервером при помощи connect(), передача структуры sockaddr_incsin_family с указанными PF_INET или PF_INET6, sin_port для указания порта прослушивания (в байтовом порядке), и sin_addr для указания IPv4 или IPv6 адреса прослушиваемого сервера (также в байтовом порядке).
- Взаимодействие с сервером при помощи send() irecv() или write() и read().
- Завершение соединения и сброс информации при вызове close(). Аналогично, если были какие-либо вызовы , каждый процесс должен закрыть (close()) сокет.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

На выполнение домашней работы отводится 16 академических часов (начало работы над заданием - 12 неделя, защита на 15 неделе).

Порядок выполнения:

1. Разработать приложение-сервер, предназначенное для параллельной обработки запросов и работающее в ОС Windows или *NIX.
2. Разработать приложение-клиент, которое будет подключаться к серверу с удаленных компьютеров Windows или *NIX.
3. Протестировать работоспособность приложений (подключится к серверному приложению и произвести обмен данными).
4. Оформить отчет.

ФОРМА ОТЧЕТА ПО ДОМАШНЕЙ РАБОТЕ

Отчет должен содержать:

- титульный лист;
- цели и задачи выполняемой работы;
- схему функционирования и описание протокола прикладного уровня разработанного клиент-серверного приложения;
- листинг разработанного клиент-серверного приложения с комментариями в местах реализации требуемых функциональных задач;
- скриншоты результатов тестирования работы приложения, иллюстрирующие основные функциональные задачи;
- выводы.

Отчет на защиту предоставляется в печатном виде

ОРГАНИЗАЦИЯ ЗАЩИТЫ ДОМАШНЕЙ РАБОТЫ

За выполнение и защиту домашнего задания максимальная оценка составляет 12 баллов, минимальная - 10 баллов. Выдача домашнего задания – 12 неделя, сдача – 15 неделя.

Оценка является суммой двух составляющих: за *своевременность* выполнения и *качество* выполнения домашнего задания.

Оценка своевременности выполнения и защиты домашнего задания баллами производится по следующим критериям:

Баллы	Критерии
2	сдача и защита домашнего задания в сроки, установленные в учебном графике
1	сдача и защита домашнего задания с отставанием не более чем на одну неделю от сроков, установленных в учебном графике
0	сдача и защита домашнего задания с опозданием более одной недели от сроков, установленных в учебном графике

Оценка качества выполнения и защиты домашнего задания баллами производится по следующим критериям:

Баллы	Критерии
10	1) уровень выполнения отвечает всем требованиям, поставленные вопросы освещены полностью, 2) высокое качество оформления, 3) четкие и правильные ответы на вопросы преподавателя при защите работы
9	1) уровень выполнения в основном отвечает требованиям, поставленные вопросы освещены полностью, 2) хорошее качество оформления, 3) в основном правильные ответы на вопросы преподавателя при защите работы
8	1) уровень выполнения в основном отвечает требованиям, поставленные вопросы освещены не полностью, 2) удовлетворительное качество оформления, 3) наличие как правильных, так и неправильных ответов на вопросы преподавателя при защите работы

0	1) уровень выполнения в основном не отвечает требованиям, поставленные вопросы освещены не полностью, 2) неудовлетворительное качество оформления, 3) в основном неправильные ответы на вопросы преподавателя при защите работы

Типовые вопросы и задания для защиты:

Оценка знаний

– Раскройте информационные источники, использованные при выполнении домашней работы.

– Раскройте понятие сокет.

Оценка умений

– Раскройте методику разработки клиент-серверных приложений на примере выполненной домашней работы.

– Опишите особенности функциональных возможностей использованных типов сокетов.

Оценка навыков

– Обоснуйте выбор типов сокетов для реализации разработанных в домашней работе приложений.

– Постройте алгоритмы работы созданных вами приложений.

ПРИЛОЖЕНИЕ А

Пример оформления листинга

Приложение сервер. Модуль «server»

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <strings.h>

#define SOMEPORT 20010
#define BUFSIZE 1024

int listen_socket, data_socket;
int addr_size;
struct sockaddr_in local_addr, remote_addr;
listen_socket = socket(PF_INET, SOCK_STREAM, 0);
```

Пример описания протокола

Заголовок			Данные	Концевик
Версия (8 бит)	Размер (16 бит)	…	Макс 1024 байт	…

Поле «версия» указывает версию протокола и может принимать значения 1 или 2. Версия 1 …

…

ПРИЛОЖЕНИЕ Б

Оформление титульного листа

Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _ *ИУК «Информатика и управление»*_____

КАФЕДРА __*ИУК5 «Системы обработки информации»*_____

ДОМАШНЯЯ РАБОТА

«Разработка сетевых приложений»

ДИСЦИПЛИНА: «Сети и телекоммуникации»

Выполнил: студент гр. _____ _____ (_____)
(Подпись) (Ф.И.О.)

Проверил: _____ (_____)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга , 202_

ЛИТЕРАТУРА

1. Э.Таненбаум, Д.Уэзеролл Компьютерные сети. 5-е изд. СПб.: Питер, 2012.

2. Е.В. Смирнова, А.В. Пролетарский, Е.А. Ромашкина и др. Технологии коммутации и маршрутизации в локальных компьютерных сетях. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2013.

3. Ачилов Р.Н. Построение защищенных корпоративных сетей. М:ДМК-пресс, 2013. https://e.lanbook.com/book/66472?category_pk=1547#book_name

Евгений Владимирович Вершинин
МаксимЕвгеньевичСмирнов

РАЗРАБОТКА СЕТЕВЫХ ПРИЛОЖЕНИЙ

Методические указания по выполнению
домашней работы по курсу "Сети и телекоммуникации"