Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Московский институт электронной техники»

Направление подготовки: 09.03.04 «Программная инженерия»

Дисциплина «Сети и телекоммуникации»

Отчёт

по лабораторной работе 5-6

Работу выполнил:

Студент группы ПИН-44

Мясников Максим Александрович

Работу проверил:

Старший преподаватель Института СПИНТех  
Виталий Владимирович Кокин

Оглавление

[Цель работы 3](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994128)

[Аннотация 3](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994129)

[Выбор языка программирования 4](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994130)

[Выбор среды разработки 5](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994131)

[Алгоритм работы приложения 6](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994132)

[Схема данных 7](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994133)

[Результат разработки 8](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994134)

[Заключение 9](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994135)

[Список использованной литературы 10](file:///C:\Users\Admin\Downloads\Seti_i_telekommunikatsii_ShABLON_OFORMLENIYa_LR.docx#_Toc81994136)

# Цель работы

Изучить принципы работы с HTTP/FTP сервером с использованием Winsock2. Изучить механизм работы с не блокирующими сокетами.

# Аннотация

В качестве протокола обмена данными с сервером можно использовать HTTP (стандарт RFC 2616), FTP (RFC 959) и другие протоколы. Их задача - доставить данные серверу и получить ответ.

HTTP — [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) передачи данных, изначально — в виде гипертекстовых документов в формате [HTML](https://ru.wikipedia.org/wiki/HTML), в настоящее время используется для передачи произвольных данных.

Основой HTTP является [технология «клиент-сервер»](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), то есть предполагается существование:

* Потребителей ([клиентов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9))), которые инициируют соединение и посылают запрос;
* Поставщиков ([серверов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5))), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

FTP — [протокол передачи](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [файлов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) по [сети](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), появившийся в 1971 году задолго до [HTTP](https://ru.wikipedia.org/wiki/HTTP) и даже до [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP), благодаря чему является одним из старейших прикладных протоколов. Изначально FTP работал поверх протокола [NCP](https://ru.wikipedia.org/wiki/Network_Control_Protocol), на сегодняшний день широко используется для распространения [ПО](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и доступа к удалённым [хостам](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82). В отличие от [TFTP](https://ru.wikipedia.org/wiki/Trivial_File_Transfer_Protocol), гарантирует передачу (либо выдачу ошибки) за счёт применения [квотируемого](https://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%BE%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) протокола.

Протокол построен на архитектуре «[клиент-сервер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)» и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. Пользователи FTP могут пройти аутентификацию, передавая логин и пароль [открытым текстом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82), или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол [SSH](https://ru.wikipedia.org/wiki/SSH) для безопасной передачи, скрывающей (шифрующей) логин и пароль, а также шифрующей содержимое.

# Выбор языка программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Требования | С++ | С# | Java | Python |
| Работа с сокетами | Да | Да | Да | Да |
| Поддержка ООП | Да | Да | Да | Да |
| Автоматическая сборка мусора | Нет | Да | Да | Да |
| Автодокументация | Нет | Да | Да | Нет |
| Реализация многопоточности | Да | Да | Да | Да |
| Обработка исключений | Да | Да | Да | Да |
| Кроссплатформенность | Да | Да (не полностью) | Да | Да |

Табл 1. Выбор языка программирования

Всем требования отвечает язык программирования Java.

# Выбор среды разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Параметры | NetBeans | IntelliJ IDEA | Eclipse |
| Тип лицензии | Бесплатно | Бесплатно (студентам) | Бесплатно |
| Наличие отладки | Да | Да | Да |
| Intellisence | Нет | Да | Нет |
| Скорость работы | Медленно | Быстро | Медленно |

Табл 2. Выбор среды разработки

Всем требованиям отвечает IntelliJ IDEA

# Алгоритм работы приложения

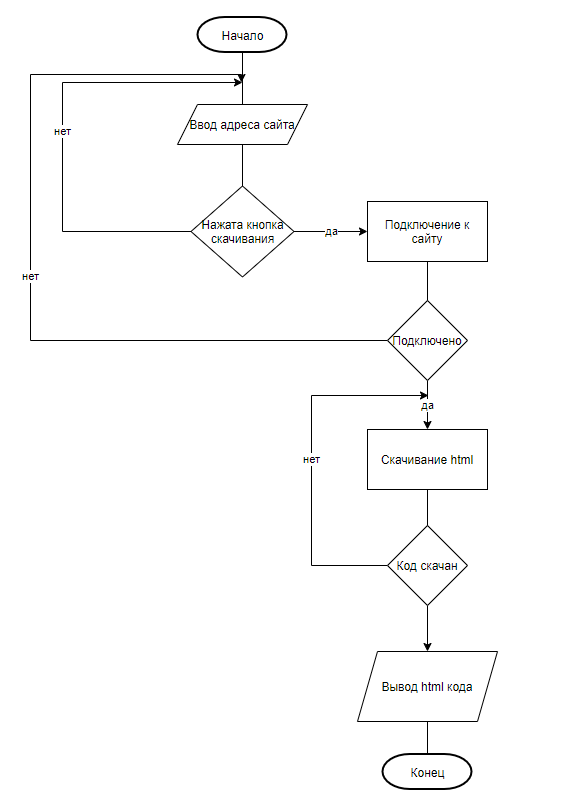


Рис 1. Алгоритм работы приложения HTTP-клиент

# Схема данных

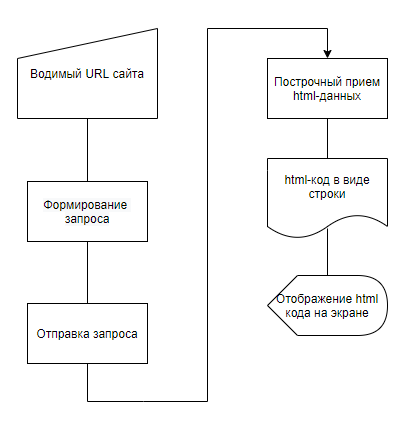


Рис 3. Схема данных HTTP-клиент

# Результат разработки

Получено 2 exe-файла: IcmpGenerator.exe и Sniffer.exe

IcmpGenerator.exe – генерирует и отправляет Icmp пакеты.

Sniffer.exe – отлавливает все пакеты проходящие через прослушиваемый адресс.

# Заключение

В процессе выполнения работы были изучены принципы работы с RAW-сокетами; изучено формирование IP, TCP, UDP, ICMP пакеты на RAW сокетах; изучены принципы приема пакетов с использованием RAW сокетов

# Список использованной литературы

1. Лабораторная работа № 5-6 Приложение для организации обмена данными c FTP/HTTP сервером. Сокеты без блокировки.
2. https://javarush.ru