# Лабораторная работа №26 Сетевое программирование сокетов

1. Цель работы
   1. Закрепить навыки работы с сетевыми сокетами на C#;
2. Литература
   1. Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВПетербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных пользователей. – Текст : электронный. – гл.15.
3. Подготовка к работе
   1. Повторить теоретический материал (см. п.2).
   2. Изучить описание лабораторной работы.
4. Основное оборудование
   1. Персональный компьютер.
5. Задание
   1. Создание клиент-серверного приложения для работы с файлами с использованием TCP-сокетов
      1. Создать новое консольное приложение-клиент. Приложение должно запрашивать у пользователя имя файла-изображения, которое необходимо отправить на сервер.

При отправке изображения на сервер сначала отправлять 4 или 8 байт с размером файла, а затем сами байты изображения

Далее клиент дожидается ответа сервера в виде 4 или 8 байт с размером файла и самих байт файла-ответа. Файл необходимо сохранить на устройстве клиента.

* + 1. Создать новое консольное приложение-сервер. Приложение должно ожидать подключения клиентов и обрабатывать все их подключения

Сервер получает размер изображения и само изображение от клиента и сжимает его размеры в 2 раза.

Для работы с изображением установить пакет System.Drawing.Common и использовать следующий код:

// imageBuffer – полученные от клиенты байты изображения

using var inputStream = new MemoryStream(imageBuffer);

using var originalImage = new Bitmap(inputStream);

int newWidth = originalImage.Width / 2;

int newHeight = originalImage.Height / 2;

using var resizedImage = new Bitmap(newWidth, newHeight);

using (var graphics = Graphics.FromImage(resizedImage))

{

graphics.InterpolationMode = InterpolationMode.HighQualityBicubic;

graphics.DrawImage(originalImage, 0, 0, newWidth, newHeight);

}

using var outputStream = new MemoryStream();

resizedImage.Save(outputStream, ImageFormat.Jpeg);

byte[] resizedBytes = outputStream.ToArray();

После сжатия сервер должен возвращать полученный файл клиенту.

* 1. Создание broadcast udp-клиента
     1. Создать консольное приложение, которое прослушивает udp-сокет на определенном порту, и с заданным интервалом передает широковещательное сообщение на тот же порт с информацией о имени игрока, текущем времени и позиции (x, y), которая случайно меняется через данный интервал. Данные можно отправлять в виде сериализованного JSON-объекта.

Одновременно с отправкой приложение должно принимать данные, которые приходят на прослушиваемый порт и отображать их на экране.

1. Порядок выполнения работы
   1. Запустить MS Visual Studio и создать оконное приложение C#.
   2. Выполнить все задания из п.5 в одном решении.
   3. Ответить на контрольные вопросы.
2. Содержание отчета
   1. Титульный лист
   2. Цель работы
   3. Ответы на контрольные вопросы
   4. Вывод
3. Контрольные вопросы
   1. Чем отличаются технологии UDP и TCP?
   2. Каков алгоритм работы сервера, использующего сокеты?