Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине: «ОСиСП»

Выполнил:

Студент 3 курса

Группы ПО-4(1)

Ковальчук В.В.

Проверил:

Дряпко А.В.

Брест 2021

Лабораторная работа № 4

Вариант 12:

**Цель**: приобрести практические навыки проектирования и разработки приложений с графическим пользовательским интерфейсом в ОС Windows.

**Задание:** Разработать сетевую утилиту для автоматического обновления приложения, разработанного в

лабораторных работах 1-3. Утилита может иметь произвольный интерфейс, определяемый ее функциональными особенностями.

2) Программа должна состоять из двух взаимодействующих частей – клиентской, устанавливаемой на компьютере с обновляемым приложением и серверной, выполняющейся на любом компьютере в локальной либо глобальной сети.

3) Клиентская часть осуществляет соединение с сервером и проверку обновлений для приложения. При наличии обновлений, все необходимые файлы загружаются и копируются в директорию с целевым приложением. В противном случае выдается соответствующее сообщение. Обработать возможные исключительные ситуации (отсутствие соединения с сервером).

4) Внести изменения в исходный проект приложения с учетом специфики загружаемых обновлений (например, хранение структуры уровня для игрового приложения в отдельном файле). То есть обновляемые ресурсы должны быть отделены от основного приложения.

5) Обновляемые компоненты по вариантам (DLL из лабораторной работы N3):

новый уровень, конфигурационный файл (скорость)

**Код**:

**private void Send()**

{

string IPAddress = "127.0.0.1";

int Port = 9000;

string Filename = @"D:\\UpdateFrom\\Update.zip";

int bufferSize = 1024;

byte[] buffer = null;

byte[] header = null;

FileStream fs = new FileStream(Filename, FileMode.Open);

bool read = true;

int bufferCount = Convert.ToInt32(Math.Ceiling((double)fs.Length / (double)bufferSize));

TcpClient tcpClient = new TcpClient(IPAddress, Port);

tcpClient.SendTimeout = 600000;

tcpClient.ReceiveTimeout = 600000;

string headerStr = "Content-length:" + fs.Length.ToString() + "\r\nFilename:" + @"D:\\UpdateFrom" + "Update.rar\r\n";

header = new byte[bufferSize];

Array.Copy(Encoding.ASCII.GetBytes(headerStr), header, Encoding.ASCII.GetBytes(headerStr).Length);

tcpClient.Client.Send(header);

for (int i = 0; i < bufferCount; i++)

{

buffer = new byte[bufferSize];

int size = fs.Read(buffer, 0, bufferSize);

tcpClient.Client.Send(buffer, size, SocketFlags.Partial);

}

tcpClient.Client.Close();

fs.Close();

}

**public void ReciveData()**

    {

        int Port = 9000;

        TcpListener listener = new TcpListener(IPAddress.Any, Port);

        listener.Start();

        Socket socket = listener.AcceptSocket();

        int bufferSize = 1024;

        byte[] buffer = null;

        byte[] header = null;

        string headerStr = "";

        string filename = "";

        int filesize = 0;

        header = new byte[bufferSize];

        socket.Receive(header);

        headerStr = Encoding.ASCII.GetString(header);

        string[] splitted = headerStr.Split(new string[] { "\r\n" }, StringSplitOptions.None);

        Dictionary<string, string> headers = new Dictionary<string, string>();

        foreach (string s in splitted)

        {

            if (s.Contains(":"))

            {

                headers.Add(s.Substring(0, s.IndexOf(":")), s.Substring(s.IndexOf(":") + 1));

            }

        }

        //Get filesize from header

        filesize = Convert.ToInt32(headers["Content-length"]);

        //Get filename from header

        filename = headers["Filename"];

        int bufferCount = Convert.ToInt32(Math.Ceiling((double)filesize / (double)bufferSize));

        FileStream fs = new FileStream(filename, FileMode.OpenOrCreate);

        while (filesize > 0)

        {

            buffer = new byte[bufferSize];

            int size = socket.Receive(buffer, SocketFlags.Partial);

            fs.Write(buffer, 0, size);

            filesize -= size;

        }

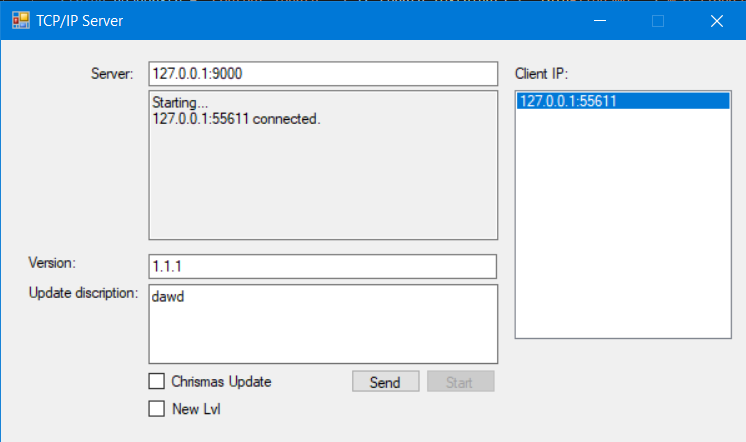
        UnZip();

        fs.Close();

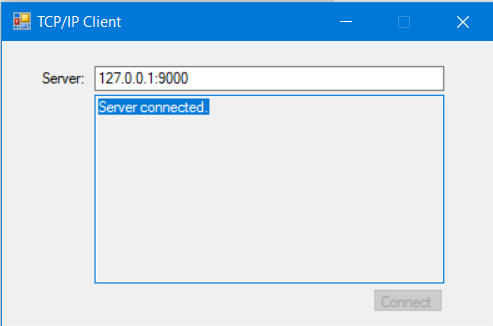
    }

**Результат**:

Окно сервера:



Окно клиента:



После запуска сервера и подключения клиента к серверу мы отправляем обновление.



В игре получаем информацию об обновлении и можем выбрать принимать обновление или нет.



Обновление применилось и теперь при нажатии на проверку обновлений нам выводит сообщение о том что обновлений нет.



**Вывод:** приобрели практические навыки проектирования и разработки приложений с графическим пользовательским интерфейсом в ОС Windows.