Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

за 5 семестр

По дисциплине: «ОСиСП»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Иваненко И. Л.

Проверил:

Дряпко А. В.

2021

1) 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).

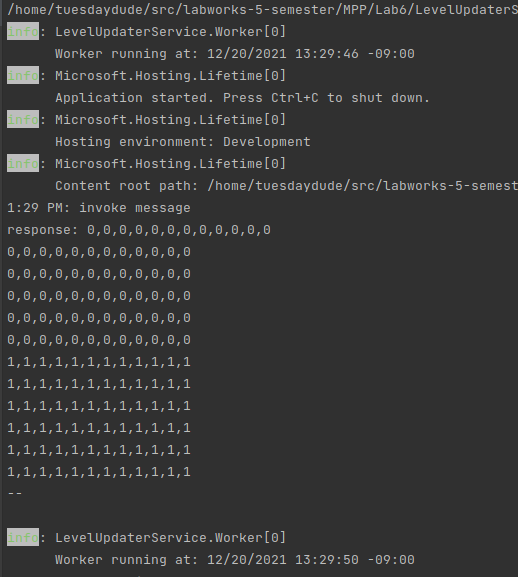
4) Внести изменения в исходный проект приложения с учетом специфики загружаемых обновлений (например, хранение структуры уровня для игрового приложения в отдельном файле). То есть обновляемые ресурсы должны быть отделены от основного приложения.

Код программы:

using System**;**  
using System**.**Net**;**  
using System**.**Net**.**Sockets**;**  
using System**.**Text**;**  
  
namespace LevelInstaller  
**{**  
public class Installer  
 **{**  
private const int **Port** = 8005**;**   
private const string **Address** = "127.0.0.1"**;**  
  
public void Update**()**  
 **{**  
var levelData = ""**;**  
try  
 **{**  
var ipPoint = new IPEndPoint**(**IPAddress**.**Parse**(Address), Port);**  
  
var socket = new Socket**(**AddressFamily**.InterNetwork,** SocketType**.Stream,** ProtocolType**.Tcp);**  
socket**.**Connect**(**ipPoint**);**  
const string **message** = "invoke message"**;**  
var data = Encoding**.**Unicode**.**GetBytes**(message**!**);**  
socket**.**Send**(**data**);**  
  
data = new byte**[**256**];**   
var builder = new StringBuilder**();**  
  
do  
 **{**  
var bytes = socket**.**Receive**(**data**,** data**.**Length**,** 0**);**  
builder**.**Append**(**Encoding**.**Unicode**.**GetString**(**data**,** 0**,** bytes**));**  
 **}**  
while **(**socket**.**Available > 0**);**  
Console**.**WriteLine**(**"response: " + builder**);**  
levelData = builder**.**ToString**();**  
  
socket**.**Shutdown**(**SocketShutdown**.Both);**  
socket**.**Close**();**  
 **}**  
catch**(**Exception ex**)**  
 **{**  
Console**.**WriteLine**(**ex**.**Message**);**  
 **}**  
  
var level = new Level**(**levelData**);**  
var installer = new LevelInstaller**(**level**);**  
installer**.**InstallLevel**();**  
 **}**  
 **}**  
**}**

uusing System**;**  
using System**.**IO**;**  
using System**.**Net**;**  
using System**.**Net**.**Sockets**;**  
using System**.**Text**;**  
  
namespace WebServer  
**{**  
public class Server  
 **{**  
private const int Port = 8005**;**  
private const string LevelDataPath =  
 @"/home/tuesdaydude/src/labworks-5-semester/MPP/Lab6/WebServer/WebServer/level.txt"**;**  
  
public void Execute**()**  
 **{**  
var ipPoint = new IPEndPoint**(**IPAddress**.**Parse**(**"127.0.0.1"**),** Port**);**  
  
var listenSocket = new Socket**(**AddressFamily**.**InterNetwork**,** SocketType**.**Stream**,** ProtocolType**.**Tcp**);**  
try  
 **{**  
listenSocket**.**Bind**(**ipPoint**);**  
  
listenSocket**.**Listen**(**10**);**  
  
while **(**true**)**  
 **{**  
var handler = listenSocket**.**Accept**();**  
var builder = new StringBuilder**();**  
var data = new byte**[**256**];**   
  
do  
 **{**  
var bytes = handler**.**Receive**(**data**);**  
builder**.**Append**(**Encoding**.**Unicode**.**GetString**(**data**,** 0**,** bytes**));**  
 **}**  
while **(**handler**.**Available>0**);**  
  
Console**.**WriteLine**(**DateTime**.**Now**.**ToShortTimeString**()** + ": " + builder**);**  
  
var levelData = File**.**ReadAllText**(**LevelDataPath**);**  
data = Encoding**.**Unicode**.**GetBytes**(**levelData**);**  
handler**.**Send**(**data**);**  
File**.**Create**(**LevelDataPath**).**Close**();**  
  
handler**.**Shutdown**(**SocketShutdown**.**Both**);**  
handler**.**Close**();**  
 **}**  
 **}**  
catch**(**Exception ex**)**  
 **{**  
Console**.**WriteLine**(**ex**.**Message**);**  
 **}**  
 **}**  
 **}**  
**}**

Результат выполнения:



Вывод: познакомился с клиент-серверной архитектурой.