Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

за 5 семестр

По дисциплине: «ОСиСП»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Иваненко И. Л.

Проверил:

Дряпко А. В.

2021

1) Основное задание заключается в доработке функционала обновления, разработка которого производилась в ЛР No4. Нужно интегрировать указанную функцию в само приложение, без использования стороннего клиента. При этом серверная часть приложения остается без изменений (возможны некоторые доработки сервера, без изменения общей клиент-серверной архитектуры);

2) Проверка обновления должна осуществляться автоматически по таймеру (QTimer) либо по непосредственному запросу пользователя. Предусмотреть выбор из меню политики обновления (с пользовательским подтверждением, без подтверждения/автоматически);

3) Сам процесс обновления должен осуществляться с использованием отдельного потока (Qthread) с минимальной вовлечённостью пользователя;

4) Необходимо отображать прогресс обновления (для этого можно использовать строку состояния – QStatusBar);

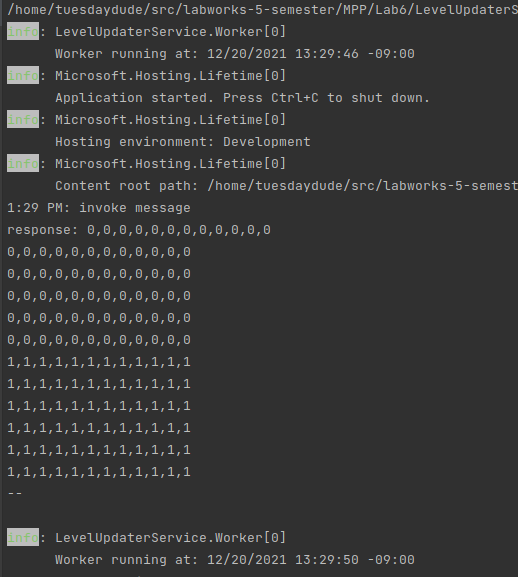
5) Для демонстрации процесса обновления и независимой работы основного и вспомогательного потоков приложения осуществлять передачу с сервера обновления помимо основных обновляемых компонентов (в соответствии с вариантом задания) одного-двух крупных файлов с произвольным содержимым

Код программы:

using LevelUpdaterService**.**Services**;**  
using Microsoft**.**Extensions**.**DependencyInjection**;**  
using Microsoft**.**Extensions**.**Hosting**;**  
  
namespace LevelUpdaterService  
**{**  
public class Program  
 **{**  
public static void Main**(**string**[]** args**)**  
 **{**  
CreateHostBuilder**(**args**).**Build**().**Run**();**  
 **}**  
  
public static IHostBuilder CreateHostBuilder**(**string**[]** args**)** =>  
 Host**.**CreateDefaultBuilder**(**args**)**  
 **.**ConfigureServices**((**hostContext**,** services**)** =>  
 **{**  
services**.**AddHostedService<Worker>**();**  
services**.**AddScoped<IClientService**,** ClientService>**();**  
services**.**AddScoped<IServerService**,** ServerService>**();**  
 **});**  
 **}**  
**}**

using System**;**  
using System**.**Threading**;**  
using System**.**Threading**.**Tasks**;**  
using LevelUpdaterService**.**Services**;**  
using Microsoft**.**Extensions**.**DependencyInjection**;**  
using Microsoft**.**Extensions**.**Hosting**;**  
using Microsoft**.**Extensions**.**Logging**;**  
  
namespace LevelUpdaterService  
**{**  
public class Worker : BackgroundService  
 **{**  
private readonly ILogger<Worker> \_logger**;**  
private readonly IServiceScopeFactory \_serviceScopeFactory**;**  
  
public Worker**(**ILogger<Worker> logger**,** IServiceScopeFactory serviceScopeFactory**)**  
 **{**  
\_logger = logger**;**  
\_serviceScopeFactory = serviceScopeFactory**;**  
 **}**  
  
protected override async Task ExecuteAsync**(**CancellationToken stoppingToken**)**  
 **{**  
var scope = \_serviceScopeFactory**.**CreateScope**();**  
var server = scope**.**ServiceProvider**.**GetService<IServerService>**();**  
var client = scope**.**ServiceProvider**.**GetService<IClientService>**();**  
while **(**!stoppingToken**.**IsCancellationRequested**)**  
 **{**  
\_logger**.**LogInformation**(**"Worker running at: {time}"**,** DateTimeOffset**.**Now**);**  
var serverThread = new Thread**(**server!**.**Up**);**  
serverThread**.**Start**();**  
Thread**.**Sleep**(**3000**);**  
var clientThread = new Thread**(**client!**.**Update**);**  
clientThread**.**Start**();**  
await Task**.**Delay**(**1000**,** stoppingToken**);**  
 **}**  
 **}**  
 **}**  
**}**

Результат выполнения:



Вывод: познакомился с клиент-серверной архитектурой.