Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

з виконаної лабораторної роботи № 7

Дисципліна: Системне програмування

на тему

«Основи використання служб ОС Windows»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

КонстантиноваЛ.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2018

**Мета:** Навчитися організовувати програми та коректно розв'язувати складні задачі. Отримати практичні навики в використанні функцій Win API.

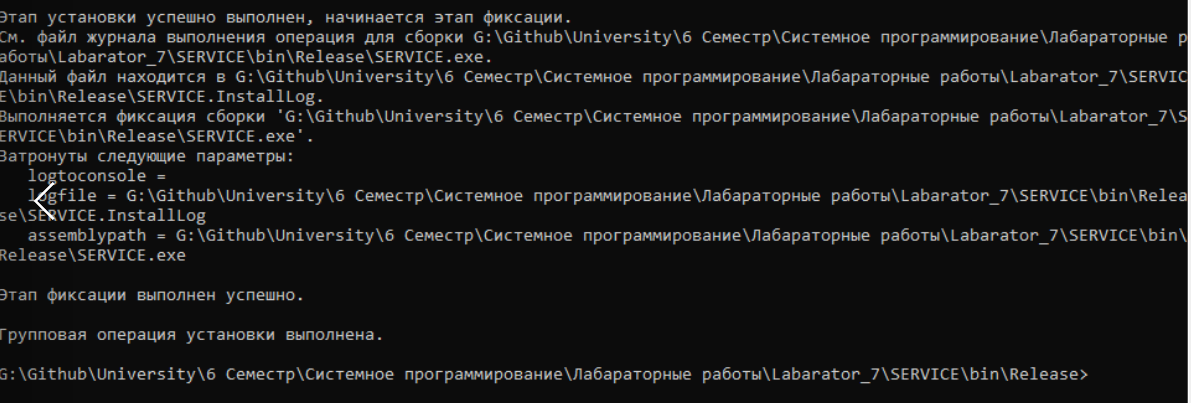
**Завдання:**

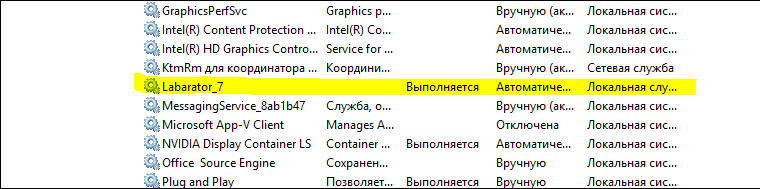
Використовуючи наведені приклади – "GetServerInfoVCL", "Service GetServerInfoSocket", "ApplicationA" та наявну електронну документацію по службам ОС WINDOWS, написати службу яка з інтервалом 5 секунд передає у мережу наступні дані:

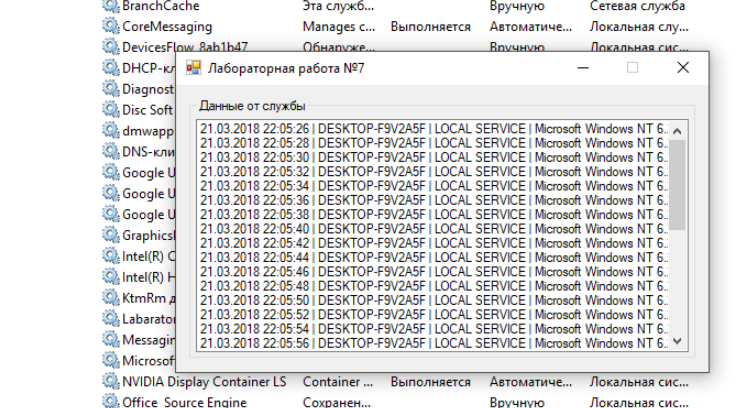
– Ім’я сервера (GetComputerName);

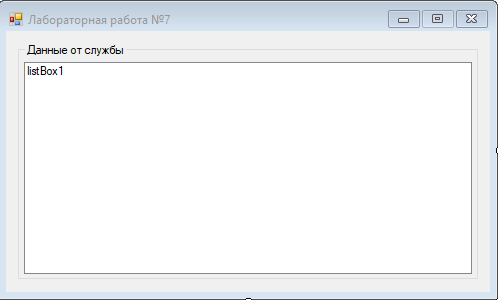
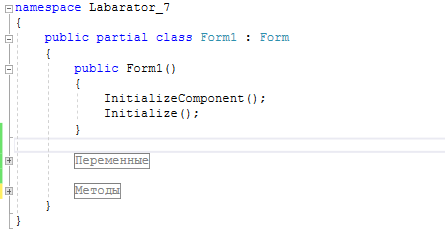
– Ім’я поточного користувача ПК (GetUserName);

– Назва та версія ОС(GetVersionEx).







SERVER 

#region Переменные

private TcpListener listener;

private TcpClient handler;

private Thread worker;

public bool IsWork { get; private set; }

public string Message { get; private set; }

#endregion

#region Методы

/// <summary>

/// Инициализация

/// </summary>

private void Initialize() {

listener = new TcpListener(new IPEndPoint(IPAddress.Parse("127.0.0.1"), 4000));

listener.Start();

worker = new Thread(Receiver);

worker.IsBackground = true;

worker.Start();

IsWork = true;

}

/// <summary>

/// Cброс

/// </summary>

private void Disconnect() {

handler.Close();

handler = null;

IsWork = false;

worker.Join(200);

}

/// <summary>

/// Прием информации в отдельном потоке

/// </summary>

private void Receiver() {

try{

byte[] data = new byte[200];

while (IsWork) {

handler = listener.AcceptTcpClient();

NetworkStream stream = handler.GetStream();

StringBuilder builder = new StringBuilder();

do {

var bytes = stream.Read(data, 0, data.Length);

builder.Append(Encoding.Unicode.GetString(data, 0, bytes));

} while (stream.DataAvailable);

var d = builder.ToString();

if (!string.IsNullOrEmpty(d))

listBox1.Invoke((MethodInvoker)delegate (){

listBox1.Items.Add(DateTime.Now + " | " + d + Environment.NewLine); });

handler.Close();

stream.Close();

Thread.Sleep(100);

}

Disconnect();

}

catch (Exception ex) {

MessageBox.Show(ex.Message);

Disconnect();

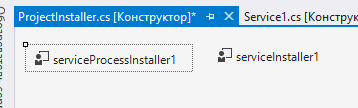
}

}

#endregion

SERVICE

static class Program

 {

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

static void Main()

{

#if DEBUG

Service1 service = new Service1();

service.OnDebug();

Thread.Sleep(System.Threading.Timeout.Infinite);

#else

ServiceBase[] ServicesToRun;

ServicesToRun = new ServiceBase[]

{

new Service1()

};

ServiceBase.Run(ServicesToRun);

#endif

}

private System.Timers.Timer timer;

private Socket socket;

/// <summary>

/// Отладочный крючок

/// </summary>

public void OnDebug() {

OnStart(null);

}

/// <summary>

/// Запуск службы

/// </summary>

/// <param name="args"></param>

protected override void OnStart(string[] args) {

try{

timer = new System.Timers.Timer(2000);

timer.Elapsed += OnTimerElapsed;

timer.Start();

}

catch (Exception ex){

OnStop();

MessageBox.Show(ex.Message);

}

}

/// <summary>

/// Остановки службы

/// </summary>

protected override void OnStop(){

try {

timer.Stop();

timer.Elapsed -= OnTimerElapsed;

timer = null;

}

catch (Exception ex) {

MessageBox.Show(ex.Message);

}

}

/// <summary>

/// Обработчик таймера

/// </summary>

private void OnTimerElapsed(object sender, ElapsedEventArgs e){

try {

IPEndPoint endPoint = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 4000);

socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

socket.Connect(endPoint);

var info = Environment.MachineName + " | " +

Environment.UserName + " | " +

Environment.OSVersion;

socket.Send(Encoding.Unicode.GetBytes(info));

socket.Shutdown(SocketShutdown.Both);

socket.Close();

socket = null;

}

catch (Exception ex) {

OnStop();

MessageBox.Show(ex.Message);

}

}