**Содержание**

1. Цель лабораторной работы 3

2. Задание 3

3. Поэтапное выполнение лабораторной работы 3

4. Листинг программы 4

Вывод 6

Список использованных источников 7

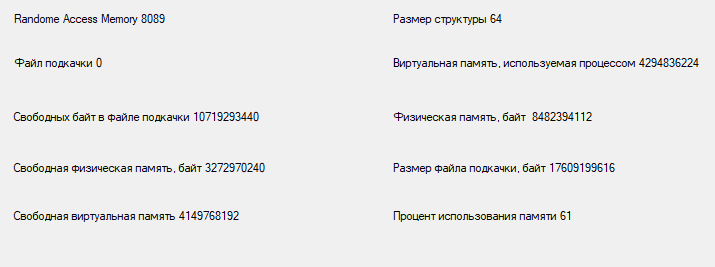
**1. Цель лабораторной работы**

Получение практических навыков по использованию Windows API для исследования памяти Windows.

**2. Задание**

1. Изучить особенности работы памяти и ее получения с использованием библиотеки С#.
2. Последовательно написать и отладить программу.
3. Выполнить задание по выведению списка данных :
4. Подготовить отчет по выполненной работе.

**3. Поэтапное выполнение лабораторной работы**1.Скачивается и устанавливается Visual Studio (В данном случае Enterprise)  
2. Устанавливается .Net FrameWork  
3. Создается проект Windows Forms   
4. Реализуется графический интерфейс с лейблами, которые принимают значение параметров (рис 1.)

  
Рисунок 1 — Графический интерфейс

5. Компиляция и запуск.

**4. Листинг программы.  
Form1.cs**using System.Windows.Forms;  
namespace ShowingSysPars{  
 public partial class Form1 : Form{  
 private ulong ram;  
 private ulong availExtendedVirtual;  
 private ulong availPageFile;  
 private ulong availPhys;  
 private ulong availVirtual;  
 public Form1(ulong ram, ulong availExtendedVirtual,   
 ulong availPageFile,   
 ulong availPhys, ulong availVirtual,  
 uint memoryLoad, ulong totalPageFile,   
ulong totalPhys, ulong totalVirtual, uint dwLength){   
 this.ram = ram;  
 this.availExtendedVirtual = availExtendedVirtual;  
 this.availPageFile = availPageFile;  
 this.availPhys = availPhys;  
 this.availVirtual = availVirtual;  
 InitializeComponent();  
label1.Text="Randome Access Memory " + ram / 1024 / 1024;  
label2.Text="Файл подкачки"+availExtendedVirtual;  
label3.Text="Свободных байт в файле подкачки"+ availPageFile;  
label4.Text="Свободная физическая память, байт"+ availPhys;  
label5.Text="Свободная виртуальная память " + availVirtual;  
label6.Text="Процент использования памяти " + memoryLoad;  
label7.Text="Размер файла подкачки, байт " + totalPageFile;  
label8.Text="Физическая память, байт " + totalPhys;  
label9.Text="Виртуальная память, используемая процессом " + totalVirtual;  
label10.Text="Размер структуры " + dwLength ;  
/\*  
DWORD dwLength; Размер структуры  
DWORD dwMemoryLoad; Процент использования памяти DWORD dwTotalPhys; Физическая память, байт  
DWORD dwAvailPhys; Свободная физическая память, байт  
DWORD dwTotalPageFile; Размер файла подкачки, байт  
DWORD dwAvailPageFile; Свободных байт в файле подкачки  
DWORD dwTotalVirtual; Виртуальная память, используемая процессом   
DWORD dwAvailVirtual; Свободная виртуальная память  
\*/  
**Program.cs**  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
namespace ShowingSysPars{  
 using System;  
 using System.Runtime.InteropServices;  
 using System.Security;  
 namespace ConsoleApplication39{  
[StructLayout(LayoutKind.Sequential,CharSet= CharSet.Auto)]  
public class MemoryStatus{  
[return: MarshalAs(UnmanagedType.Bool)]  
[DllImport("kernel32.dll",CharSet=CharSet.Auto, SetLastError = true)]  
internal static extern bool GlobalMemoryStatusEx([In, Out] MemoryStatus lpBuffer);  
 public uint dwLength;  
 public uint MemoryLoad;  
 public ulong TotalPhys;  
 public ulong AvailPhys;  
 public ulong TotalPageFile;  
 public ulong AvailPageFile;  
 public ulong TotalVirtual;  
 public ulong AvailVirtual;  
 public ulong AvailExtendedVirtual;  
 private static volatile MemoryStatus singleton;  
 private static readonly object syncroot = new object();  
 public static MemoryStatus CreateInstance(){  
 if (singleton == null)  
 lock (syncroot)  
 if (singleton == null)  
 singleton = new MemoryStatus();  
 return singleton; }  
 [SecurityCritical]  
 private MemoryStatus(){  
dwLength = (uint)Marshal.SizeOf(typeof(MemoryStatus));  
 GlobalMemoryStatusEx(this);}  
 class Program{  
 static void Main(){  
MemoryStatus status = MemoryStatus.CreateInstance();  
ulong ram = status.TotalPhys;  
Console.WriteLine("RAM = {0} MB", ram / 1024 / 1024); //Console.ReadKey(); Application.EnableVisualStyles();  
Application.SetCompatibleTextRenderingDefault(false);   
Application.Run(new Form1(ram,status.AvailExtendedVirtual,  
 status.AvailPageFile, status.AvailPhys,  
 status.AvailVirtual, status.MemoryLoad ,  
 status.TotalPageFile,status.TotalPhys,  
 status.TotalVirtual,status.dwLength ));}}}}  
Остальные файлы были получены в результате генерации Visual Studio.

**Вывод**

Получены практические навыки по использованию Windows API для исследования памяти Windows.

**Список использованных источников**

1. Чеповский, А.М. Common Intermediate Language и системное программирование в Microsoft .NET [Электронный ресурс] : учебное пособие / А.М. Чеповский, А.В. Макаров, С.Ю. Скоробогатов. — Электрон. дан. — Москва : , 2016. — 398 с. — Режим доступа: https://e.lanbook.com/book/100589. — Загл. с экрана.

2. Голдштейн, С. Оптимизация приложений на платформе .NET [Электронный ресурс] / С. Голдштейн, Д. Зурбалев, И. Флатов ; пер. с англ. Киселев А.Н.. — Электрон. дан. — Москва : ДМК Пресс, 2014. — 524 с. — Режим доступа: https://e.lanbook.com/book/93266. — Загл. с экрана.

3.Чеповский, А.М. Common Intermediate Language и системное программирование в Microsoft .NET [Электронный ресурс] : учебное пособие / А.М. Чеповский, А.В. Макаров, С.Ю. Скоробогатов. — Электрон. дан. — Москва : , 2016. — 398 с. — Режим доступа: https://e.lanbook.com/book/100589. — Загл. с экрана.

4. Крищенко, В.А. Поддержка разработки распределенных приложений в Microsoft .NET Framework [Электронный ресурс] : учебное пособие / В.А. Крищенко, С.В. Горин. — Электрон. дан. — Москва : , 2016. — 249 с. — Режим доступа: https://e.lanbook.com/book/100367. — Загл. с экрана.

5. Суханов, М.В. Основы Microsoft .NET Framework и языка программирования C# [Электронный ресурс] : учебное пособие / М.В. Суханов, И.В. Бачурин, И.С. Майров. — Электрон. дан. — Архангельск : САФУ, 2014. — 96 с. — Режим доступа: https://e.lanbook.com/book/96543. — Загл. с экрана.