Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Новосибирский государственный технический университет»

Кафедра систем сбора и обработки данных

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по программированию

на тему: «Менеджер задач»

Автор проекта: Панков М. Ю.

Специальность:

Информационные системы и технологии

Группа: АТ-63

Факультет: АВТФ

Руководитель: Черкашин С. В.

Воронов В. В.

Новосибирск

2017 г.

Техническое Задание:

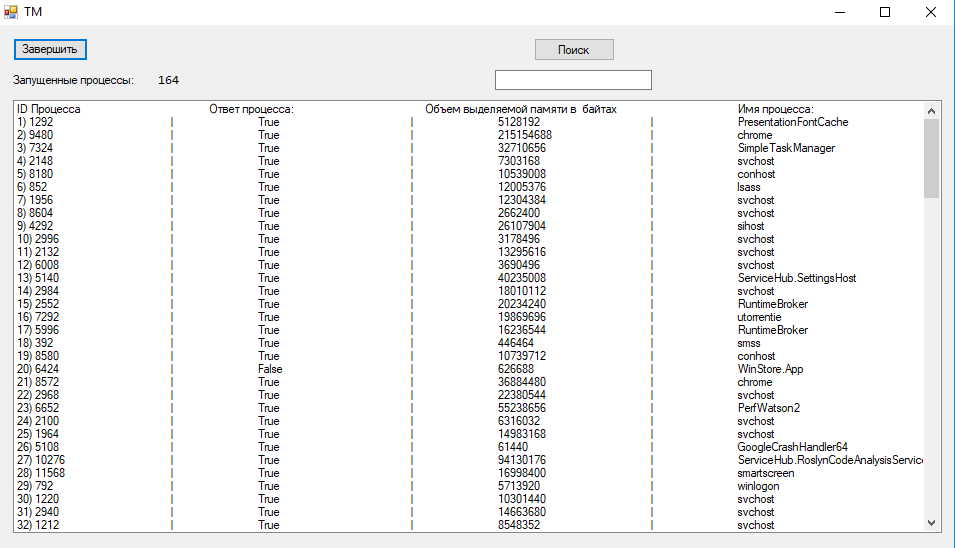
Создание Менеджера задач, с приведенными ниже функциями и информацией.

**Функции**

1. Поиск процесса
2. Удаление процесса

Описание разрабатываемой программы

Скриншот:



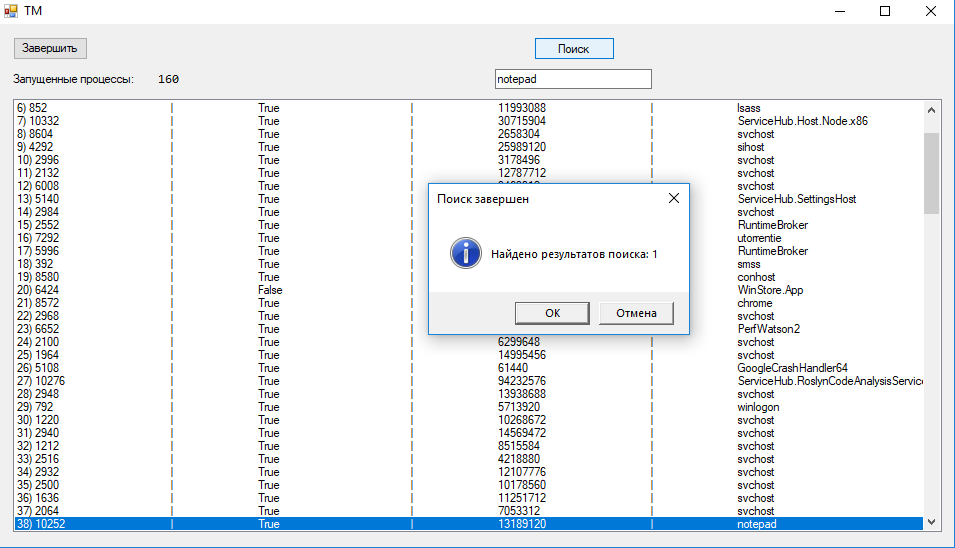
При запуске, программа показывает нам все процессы запущенные на данном компьютере, их имя, PID, объем памяти, выделяемой для каждого прцесса, и ответ процесса.

Программа, как было уже выше сказано имеет 2 функции, первая это поиск процесса, вторая это его удаление.

Поиск осуществляется по названию процесса, либо по другим данным.

Рассмотрим на примере. Я открою блокнот, в списке процессов он будет называться notepad, и на его примере покажу поиск процесса.

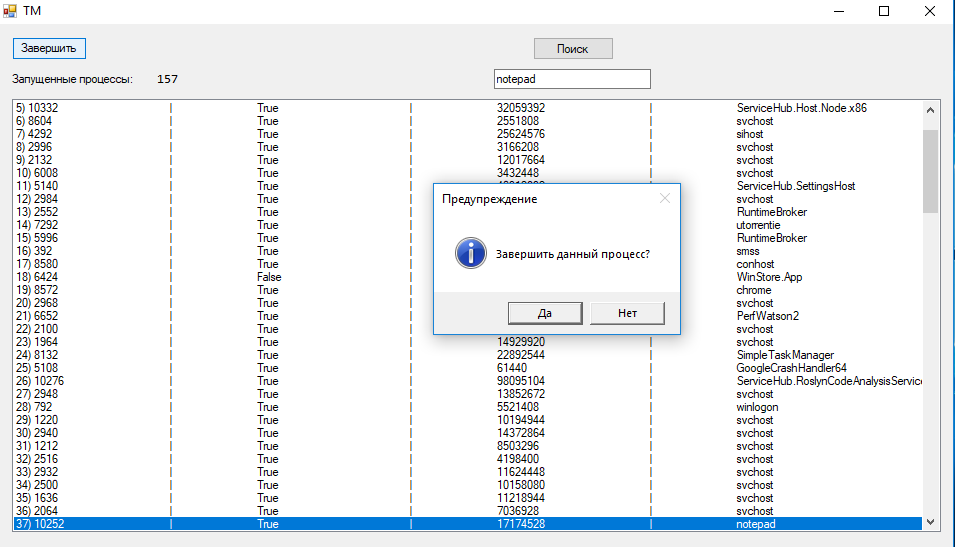
Скриншот:



Также после завершения поиска нам будет показан messagebox c счетчиком кол-ва элементов подходящих под условие поиска.

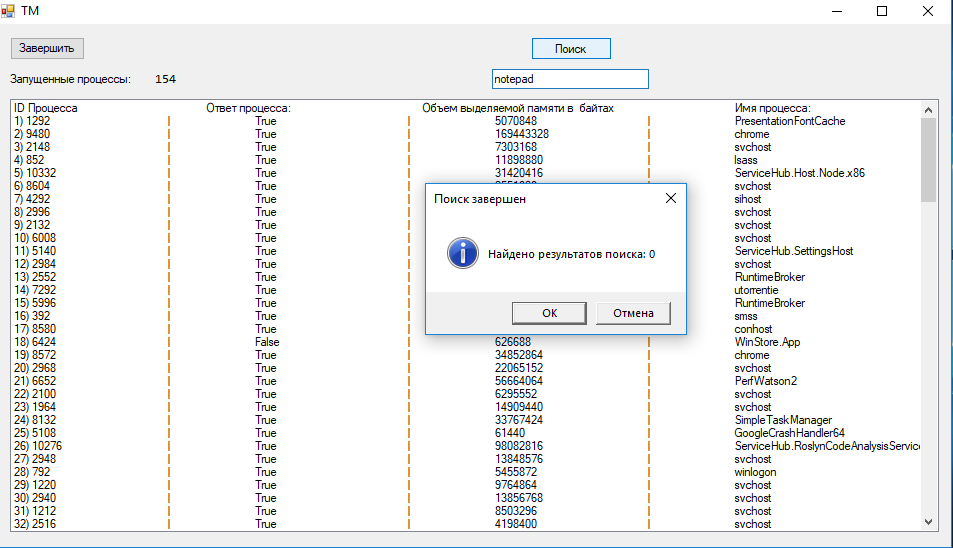
1. Удаление процесса

Удаление процесса в моей программе происходит по индексу в списке процессов. Также на примере блокнота рассмотрим удаление процесса из списка.



Тут также появляется messegebox в котором нас спрашивают закрыть данный процесс или же нет.

После нажатия “Да” процесс закрылся и это можно проверить его отсутствием в списке процессов.



Приложения

Класс камня Stone:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Diagnostics;

namespace TaskManager

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Process[] procs;

private void GetProcesses()

{

procs = Process.GetProcesses();

if (Convert.ToInt32(label2.Text) != procs.Length) // Проверяем, были ли запущены или завершены новые процессы

{

listBox1.Items.Clear();

listBox1.Items.Add("ID Процесса"+

" \t \t \t Ответ процесса: " +

" \t \t \t Объем выделяемой памяти в " + " байтах" +

" \t \t \t Имя процесса: ");

for (int i = 0; i < procs.Length; i++)

{

listBox1.Items.Add((i+1)+") " + procs[i].Id +

" \t \t | \t \t " +procs[i].Responding +

" \t \t | \t \t " + procs[i].WorkingSet64 +

" \t \t | \t \t " + procs[i].ProcessName);

// Добавить имя процесса в список

//listBox1.Items.Add(procs[i].Id);

// listBox1.Items.Add(procs[i].WorkingSet64);

}

label2.Text = procs.Length.ToString();

}

}

private void Form1\_Load(object sender, EventArgs e)

{

GetProcesses();

}

// Проверяем каждую секунду изменениями в списке процессов

private void timer1\_Tick(object sender, EventArgs e)

{

GetProcesses();

}

private void button1\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show(

"Завершить данный процесс?",

"Предупреждение",

MessageBoxButtons.YesNo,

MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

if (result == DialogResult.Yes)

procs[listBox1.SelectedIndex-1].Kill(); // Завершаем процесс, соответствующий выбранному индексу в списке;

}

private void kIllProcessToolStripMenuItem\_Click(object sender, EventArgs e)

{

procs[listBox1.SelectedIndex-1].Kill();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void btnsearch\_Click(object sender, EventArgs e)

{

listBox1.SelectedItems.Clear();

for (int i = listBox1.Items.Count - 1; i >= 0;i--)

{

if (listBox1.Items[i].ToString().ToLower().Contains(txtsearch.Text.ToLower()))

listBox1.SetSelected(i, true);

}

DialogResult result = MessageBox.Show(

"Найдено результатов поиска: " + listBox1.SelectedItems.Count.ToString(),

"Поиск завершен",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

// label3.Text = "Найдено результатов поиска: "+ listBox1.SelectedItems.Count.ToString();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged\_1(object sender, EventArgs e)

{

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Diagnostics;

namespace TaskManager

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Process[] procs;

private void GetProcesses()

{

procs = Process.GetProcesses();

if (Convert.ToInt32(label2.Text) != procs.Length) // Проверяем, были ли запущены или завершены новые процессы

{

listBox1.Items.Clear();

listBox1.Items.Add("ID Процесса"+

" \t \t \t Ответ процесса: " +

" \t \t \t Объем выделяемой памяти в " + " байтах" +

" \t \t \t Имя процесса: ");

for (int i = 0; i < procs.Length; i++)

{

listBox1.Items.Add((i+1)+") " + procs[i].Id +

" \t \t | \t \t " +procs[i].Responding +

" \t \t | \t \t " + procs[i].WorkingSet64 +

" \t \t | \t \t " + procs[i].ProcessName);

// Добавить имя процесса в список

//listBox1.Items.Add(procs[i].Id);

// listBox1.Items.Add(procs[i].WorkingSet64);

}

label2.Text = procs.Length.ToString();

}

}

private void Form1\_Load(object sender, EventArgs e)

{

GetProcesses();

}

// Проверяем каждую секунду изменениями в списке процессов

private void timer1\_Tick(object sender, EventArgs e)

{

GetProcesses();

}

private void button1\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show(

"Завершить данный процесс?",

"Предупреждение",

MessageBoxButtons.YesNo,

MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

if (result == DialogResult.Yes)

procs[listBox1.SelectedIndex-1].Kill(); // Завершаем процесс, соответствующий выбранному индексу в списке;

}

private void kIllProcessToolStripMenuItem\_Click(object sender, EventArgs e)

{

procs[listBox1.SelectedIndex-1].Kill();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void btnsearch\_Click(object sender, EventArgs e)

{

listBox1.SelectedItems.Clear();

for (int i = listBox1.Items.Count - 1; i >= 0;i--)

{

if (listBox1.Items[i].ToString().ToLower().Contains(txtsearch.Text.ToLower()))

listBox1.SetSelected(i, true);

}

DialogResult result = MessageBox.Show(

"Найдено результатов поиска: " + listBox1.SelectedItems.Count.ToString(),

"Поиск завершен",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

// label3.Text = "Найдено результатов поиска: "+ listBox1.SelectedItems.Count.ToString();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged\_1(object sender, EventArgs e)

{

}

}

}