Міністерствоосвіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет Прикладної математики

Кафедра спеціалізованих комп’ютерних систем

Лабораторна робота №2

*з дисципліни:*

«Комп’ютерна архітектура»

Організація багато поточного виконання програм .NET

Варіант №16

Виконав:

студент групи КВ-92

Степанюк М.Ф.

Перевірив:

Петрашенко А. В.

Київ 2012

**Мета роботи:** вивчення технології багато поточного програмування засобами .NET Framework та мови C# на прикладі реалізації задачі інформаційного пошуку.

**Завдання:** Використовуючи середовища розробки Visual Studio .NET і мову програмування C#, реалізувати функцію пошуку фрази серед набору документів, представлених текстовими файлами.

**Задачі потоків**

Головний потік – обробка подій від користувача, виведення результатів пошуку

Потік 1 – підготовка колекції до пошуку: створення індексів документів

Потік 2 – пошук у колекції за варіантом

Потік 3 – пошук у колекції BitArray

Пошук у колекції: HashTable

Функція об’єднання слів у запиті: a AND b

Об’єкт синхронізації потоків: Mutex

***Текст програми:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Collections;

using System.Threading;

using System.IO;

using System.Windows.Forms;

namespace KA\_2

{

public partial class Form1 : Form

{

Hashtable ht;

RichTextBox RichTextBox2;

Mutex mt, mt2, mt1, mt3;

bool ext = true, pp1, butclick = false;

int len = 0, cnt, cnt\_bit, pp=0;

TimeSpan ts1, ts2;

string[,] search;

Thread th\_main, th1, th2, th3;

ListBox lst1, lst2, lst3, lst4;

public Form1()

{

InitializeComponent();

mt = new Mutex(); //for HashTasble

mt1 = new Mutex(); //for lst1, lst2

mt2 = new Mutex(); //for ListBox1

mt3 = new Mutex(); //for lst3, lst4

ts1 = new TimeSpan();

ts2 = new TimeSpan();

}

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

{

ht = new Hashtable();

lst1 = new ListBox();

lst2 = new ListBox();

lst1.Visible = false;

lst2.Visible = false;

lst3 = new ListBox();

lst4 = new ListBox();

lst3.Visible = false;

lst4.Visible = false;

}

public void button1\_Click\_1(object sender, EventArgs e)

{

if (opf.ShowDialog() == DialogResult.OK)

{

FPath.Text = opf.FileName;

mt2.WaitOne();

if (!listBox1.Items.Contains(FPath.Text))

{

fileNames.Text += listBox1.Items.Count.ToString() + "\n";

listBox1.Items.Add(FPath.Text);

}

mt2.ReleaseMutex();

if (th1 == null)

{

FPath.Text = "";

return;

}

butclick = true;

}

}

private int SearchWords( string fname )

{

for (int i = 0; i < len; i++)

if (ht[fname] != null && ((string[])ht[fname]).ToList().Contains(search[0, i]))

{

search[1, i] = "1";

}

int mul = 1;

for (int j = 0; j < len; j++)

{

mul \*= Convert.ToInt16(search[1, j]);

search[1, j] = "0";

}

return mul;

}

private void AddToHash(string fname)

{

string s = System.IO.File.ReadAllText(fname, Encoding.Default);

s = s.ToUpper();

string[] WordsArr = s.Split(',', '.', '!', '?', '-', ';', '\r', '\n', ':', '—', ' ', ')', '(', '\"', '/');

ht.Add(ht.Count.ToString(), WordsArr);

}

private void PrepareThread()

{

foreach ( string str in listBox1.Items )

{

mt.WaitOne();

AddToHash(str);

DateTime dt = DateTime.Now;

lst2.Items.Add(dt);

mt.ReleaseMutex();

}

while ( ext )

{

if ( butclick )

{

mt.WaitOne();

AddToHash(FPath.Text);

DateTime dt = DateTime.Now;

lst2.Items.Add(dt);

mt.ReleaseMutex();

FName.Text = "";

FPath.Text = "";

butclick = false;

Thread.Sleep(100);

}

Thread.Sleep(100);

}

but\_ser.Enabled = true;

}

private void SearchThread()

{

cnt = 0;

while ( ext )

{

mt.WaitOne();

if (cnt >= ht.Count)

{

mt.ReleaseMutex();

Thread.Sleep(500);

continue;

}

mt.ReleaseMutex();

DateTime dt = DateTime.Now;

mt1.WaitOne();

bool res1 = ( SearchWords(cnt.ToString()) > 0 );

lst1.Items.Add(res1);

ts1 = DateTime.Now - (DateTime)lst2.Items[cnt];

lst2.Items[cnt] = ts1;

mt1.ReleaseMutex();

Thread.Sleep(500);

cnt++;

Thread.Sleep(100);

}

but\_ser.Enabled = true;

}

private void BitSearch()

{

cnt\_bit = 0;

while (ext)

{

mt2.WaitOne();

if (cnt\_bit >= listBox1.Items.Count)

{

mt2.ReleaseMutex();

Thread.Sleep(500);

continue;

}

string fname = listBox1.Items[cnt\_bit].ToString();

mt2.ReleaseMutex();

DateTime dt = DateTime.Now;

string s = System.IO.File.ReadAllText(fname, Encoding.Default);

s = s.ToUpper();

string[] WordsArr = s.Split(',', '.', '!', '?', '-', ';', '\r', '\n', ':', '—', ' ', ')', '(', '\"', '/');

int j = 0;

bool[] resArr = new bool[len];

BitArray BA = new BitArray(resArr);

string[] unrep = new string[WordsArr.Length];

for (j = 0; j < len; j++)

{

BA[j] = (BA[j] || WordsArr.ToList().Contains(search[0, j]));

}

bool add\_buf = true;

for (j = 0; j < len; j++)

add\_buf = (BA[j] && add\_buf);

mt3.WaitOne();

lst3.Items.Add(add\_buf);

ts2 = DateTime.Now - dt;

lst4.Items.Add(ts2);

mt3.ReleaseMutex();

cnt\_bit++;

}

}

private void ResThread()

{

int i = 0, j = 0;

while (ext)

{

mt1.WaitOne();

if (i < lst1.Items.Count)

{

result\_box.Text += lst1.Items[i].ToString() + "\n";

time\_box.Text += lst2.Items[i].ToString() + "\n";

i++;

}

mt1.ReleaseMutex();

mt3.WaitOne();

if (j < lst3.Items.Count)

{

bit\_res.Text += lst3.Items[j].ToString() + "\n";

bit\_time.Text += lst4.Items[j].ToString() + "\n";

j++;

}

mt3.ReleaseMutex();

Thread.Sleep(500);

}

}

private void button2\_Click(object sender, EventArgs e)

{

button2.Visible = false;

ext = true;

th1 = new Thread(new ThreadStart(PrepareThread));

th2 = new Thread(new ThreadStart(SearchThread));

th3 = new Thread(new ThreadStart(BitSearch));

th\_main = new Thread(new ThreadStart(ResThread));

th1.Start();

th2.Start();

th3.Start();

th\_main.Start();

button2.Enabled = false;

}

private void button3\_Click(object sender, EventArgs e)

{

ext = false;

button2.Enabled = true;

th1.Abort();

th2.Abort();

th3.Abort();

th\_main.Abort();

MessageBox.Show("Виконання потоків закінчено. Програму буде автоматично закрито");

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

but\_ser.Visible = false;

button1.Visible = true;

button2.Visible = true;

button3.Visible = true;

len = 0;

srcline.Text += " ";

string[] WordsArr = srcline.Text.Split(' ', ',','\0','\n');

string[] unrep = new string[WordsArr.Length];

for (int i = 0; i < WordsArr.Length-1; i++)

WordsArr[i] = WordsArr[i].ToUpper();

for (int i = 0; i < WordsArr.Length-1; i++)

if (!unrep.ToList().Contains(WordsArr[i]))

{

unrep[len] = WordsArr[i];

len++;

}

search = new string[2, len];

for (int i = 0; i < len; i++)

{

search[0, i] = unrep[i];

search[1, i] = "0";

}

mt.WaitOne();

cnt = 0;

mt.ReleaseMutex();

mt2.WaitOne();

cnt\_bit = 0;

mt2.ReleaseMutex();

}

private void button4\_Click\_1(object sender, EventArgs e)

{

if (!pp1)

{

RichTextBox2 = new RichTextBox();

RichTextBox2.LoadFile(opf.FileName);

textBox1.Text = Convert.ToString(RichTextBox2.Find(srcline.Text));

pp1 = true;

}

}

}

}

***Результати роботи програми:***

