**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №4**

з дисципліни

«Операційні системи»

**Виконав:**

студент групи КН-208

Телішевський Петро

**Викладач:**

Кривенчук Ю.П.

Львів – 2018 р.

**Тема**

Робота з динамічними бібліотеками в ОС Windows.

**Мета**

Ознайомитися з динамічно-зв’язувальними бібліотеками (Dynamic-link library) в ОС Windows. Навчитися реалізовувати динамічно-зв’язувальні бібліотеки.

**Завдання.**

* Реалізувати лабораторну роботу №4 у вигляді динамічно-зв’язувальної бібліотеки.
* Запустити створену бібліотеку з командної стрічки (cmd.exe) за допомогою rundll32.exe.
* Створити окрему програму і реалізувати статичний зв'язок між програмою та бібліотекою.
* Створити окрему програму і реалізувати динамічний зв'язок між програмою та бібліотекою.
* Експортувати головну функцію бібліотеки під іншим іменем.
* Результати виконання роботи відобразити у звіті.

DLL бібліотека

#include "stdafx.h"

#include "Dll1.h"

using namespace std;

struct ARGV {

int \*arr;

int start;

int end;

};

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void func(int \*arr, int start, int end) {

int i, j;

for (i = start; i < end - 1; i++)

for (j = start; j < end - start - 1; j++)

if (arr[j] > arr[j + 1])

swap(&arr[j], &arr[j + 1]);

}

unsigned \_\_stdcall func2(void\* n) {

ARGV \*args = (ARGV\*)n;

func(args->arr, args->start, args->end);

return 0;

}

extern "C" \_\_declspec(dllexport) void LetterList()

{

int number;

cout << "Enter count of threads" << endl;

cin >> number;

int n;

cout << "Array length: " << endl;

cin >> n;

int \*arr = new int[n];

srand((unsigned)time(0));

for (int i = 0; i < n; i++) {

arr[i] = (rand() % 100);

}

int step = n / number;

cout << "Before swapping: " << endl;

for (int i = 0; i < n; i++) {

cout << arr[i] << " ";

if ((i + 1) % 10 == 0) {

cout << endl;

}

}

cout << endl;

HANDLE \*myhandle = new HANDLE[number];

ARGV \*args\_arr = new ARGV[number];

for (int i = 0; i < number; i++) {

if (i == number - 1) {

args\_arr[i] = { arr, i\*step, n };

}

else {

args\_arr[i] = { arr, i\*step, (i + 1) \* step };

}

}

clock\_t start = clock();

for (int i = 0; i < number; i++) {

myhandle[i] = (HANDLE)\_beginthreadex(0, 0, &func2, &args\_arr[i], 0, 0);

}

HANDLE hSem = CreateSemaphore(NULL, 0, number, NULL);

for (int i = 0; i < number; i++) {

CloseHandle(myhandle[i]);

}

double end = (double)(clock() - start) / CLOCKS\_PER\_SEC;

cout << endl;

func(arr, 0, n);

cout << endl;

cout << "After swapping" << endl;

for (int i = 0; i < n; i++) {

cout << arr[i] << " ";

if ((i + 1) % 10 == 0) {

cout << endl;

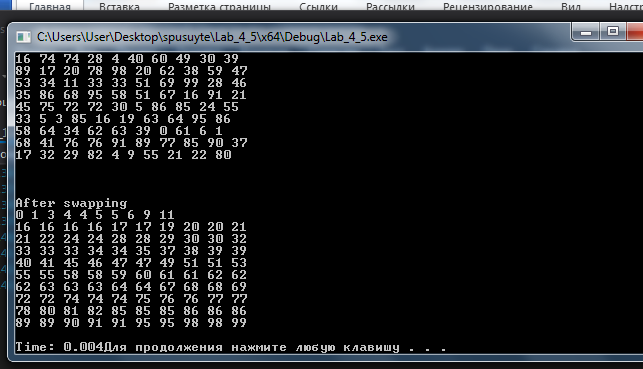
}

}

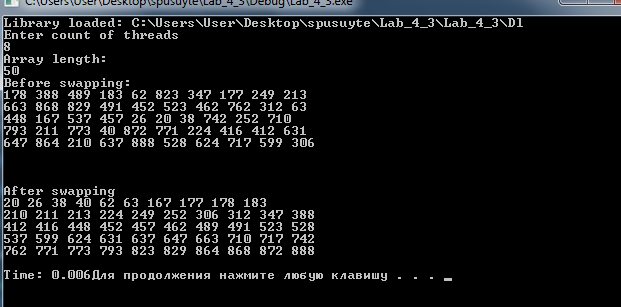
cout << endl << "Time: " << end;

}

Результат із статичним зв'язоком між програмою та бібліотекою:



Результат із динамічним зв'язоком між програмою та бібліотекою:



Висновок:

Навчився реалізовувати динамічно-зв’язувальні бібліотеки.