

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №1**

**з дисципліни «Операційні системи»**

**на тему: «Створення потоків і їх синхронізація в режимі користувача»**

Виконав:

студент гр. ПЗ2011 Кулик С. В.

Прийняв:

Андрющенко В. О.

Дніпро, 2022

**2. Короткі теоретичні відомості:**

Програмний код, якому необхідно "виділити процесор" в операційній системі Windows називається Потоком (Thread).

Критичні секції - це найпростіші об'єкти, які не знижують загальної ефективності додатка. Помітивши блок код як critical section, можна синхронізувати доступ до нього від декількох потоків. Спочатку слід оголосити глобальну структуру;

Створення потока Для створення потоку використовують функцію potok=(HANDLE)\_beginthread(адреса функції, розмір виділеної пам’яті, список параметрів);

Всі потоки та їх дані розміщуються в конкретному Процесі (Process). З точки зору прикладної програми Процес – це адресний простір, в якому розміщуються код та дані всіх EXE та DLL модулів. Саме в ньому знаходяться області пам’яті, які динамічно розподіляються для стеків потоку та інших потреб.

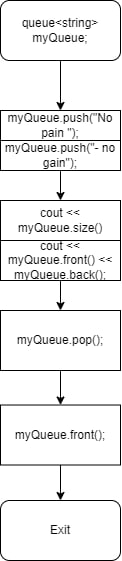
**3. Завдання:**

Розробити програму, яка створює два або більше потока керування (системний виклик CreateThread), що отримують доступ до спільних ресурсів за принципом взаємовиключного доступу. Для синхронізації застосовувати системні структури CRITICAL\_SECTION. Характеристики ресурсів і спосіб їх використання потоками відповідає індивідуальному завданню.

**3.1 Індивідуальне завдання:**

1) Черга на основі масиву текстової інформації фіксованого розміру.

**4. Схема алгоритму:**



**5. Текст програми:**

#include <windows.h>

#include <process.h>

#include <iostream>

#include <conio.h>

#include <queue>

#include <string>

using namespace std;

#define MAX\_ARRAY 5

CRITICAL\_SECTION critsect;

int array[MAX\_ARRAY];

HANDLE Empty, Full, Print1, Print2;

void EmptyArray(void\*);

void PrintArray(void\*);

void EmptyArray(void\*) {

cout << "\nThis Array is empty" << endl;

EnterCriticalSection(&critsect);

queue<string>myQueue;

cout << "Number of elements in the queue: " << myQueue.size() << endl;

Sleep(1000);

LeaveCriticalSection(&critsect);

\_endthread();

}

void PrintArray(void\*) {

cout << "\nPrint Array:" << endl;

EnterCriticalSection(&critsect);

queue<string>myQueue;

myQueue.push("No pain");

myQueue.push("- no gain");

cout << "\nHere it is: " << myQueue.front() << myQueue.back();

myQueue.pop();

cout << "\nNow there is one element left in the queue: " << myQueue.front();

}

int main()

{

InitializeCriticalSection(&critsect);

Empty = (HANDLE)\_beginthread(EmptyArray, 1024, NULL);

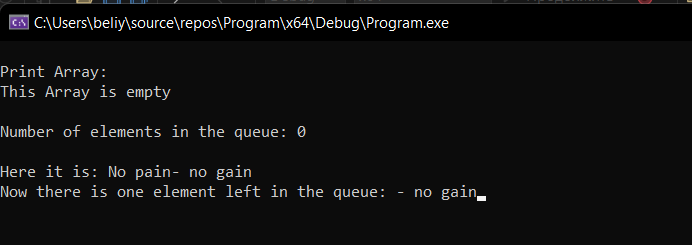
Print1 = (HANDLE)\_beginthread(PrintArray, 1024, NULL);

Sleep(10000);

DeleteCriticalSection(&critsect);

}

**6. Результат виконання програми:**



**7. Аналіз результатів:**

На фото ми можемо побачити роботу програми. Програма може видаляти елементи у черзі. Програма показує початковий текст перед редагуванням.

**8. Висновок:**

Всі потоки та їх дані розміщуються в конкретному Процесі (Process). З точки зору прикладної програми Процес – це адресний простір, в якому розміщуються код та дані всіх EXE та DLL модулів. Саме в ньому знаходяться області пам’яті, які динамічно розподіляються для стеків потоку та інших потреб. При запуску на виконання EXE програми система створює для неї Процес, в контексті якого запускає первинний Потік. Цей Потік може запустити навиконання ще дочірні Потоки, які будуть виконуватися паралельно. Коли всі Потоки знищуються, Процес теж знищується.