

Лабораторна робота № 7

(4 години)

Тема: Керування потоками у Windows.

Мета: Ознайомитися із засобами створення потоків та сформувати навички керування потоками у Windows-програмах.

Короткі теоретичні відомості

Потік (thread) – незалежна одиниця виконання програмного коду в межах процесу. Потік існує в адресному просторі процесу та одержує доступ до всіх його ресурсів. Кожному потоку надаються регістри процесора, стек, та локальна область зберігання (TLS), які разом складають контекст потоку.

Кожний процес має принаймні один (первинний) потік, який представлений головною функцією (**main()**, **WinMain()** та ін). Додаткові потоки створюються за допомогою системного виклику **CreateThread()**.

HANDLE CreateThread(

```
LPSECURITY_ATTRIBUTES lpThreadAttributes, // вказівник на TSA (або NULL)
DWORD dwStackSize,                       // розмір стеку потоку в байтах
                                           // (або 0 – за умовчанням 1MB)
LPTHREAD_START_ROUTINE lpStartAddress, // вказівник на функцію потоку
LPVOID lpParameter,                    // параметр для потоку
DWORD dwCreationFlags,                  // прапорці створення
LPDWORD lpThreadId                       // вказівник на ідентифікатор потоку,
                                           // (або NULL)
```

);

Прапорці створення *dwCreationFlags* визначають стан потоку після запуску (0 – потік запускається відразу, **CREATE_SUSPENDED** – потік запускається у призупиненому стані, із якого виходить по **ResumeThread()**).

Функція повертає дескриптор потоку та його ідентифікатор (через *lpThreadId*). Якщо при виклику функції задати *lpThreadId* рівним NULL, то ідентифікатор не повертається.

Функція потоку приймає один безтиповий аргумент (найчастіше вказівник) і повертає значення **DWORD**. Вона може мати будь-яке ім'я:

```
DWORD WINAPI ThreadFunc(LPVOID lpParam)
{
    ...
    ExitThread (dwExitCode); // або return ExitCode;
}
```

Закінчується функція потоку функцією **ExitThread()** або оператором **return**, які повертають первинному потоку код завершення.

Завершення потоку

Виконання потоку може бути завершено одним із способів:

- функція потоку завершується і повертає керування (бажаний спосіб);
- викликається функція **ExitThread()**;
- викликається функція **TerminateThread()** (небажаний спосіб);
- завершується процес, який містить потік (небажаний спосіб).

За допомогою функції **GetExitCodeThread()** можна одержати код завершення потоку, знаючи його дескриптор.

BOOL GetExitCodeThread(

```
HANDLE hThread, // дескриптор потоку
LPDWORD lpExitCode // адреса змінної для одержання коду
);
```

Якщо потік не завершив свого виконання, функція повертає значення **STILL_ACTIVE**.

Потоки можуть бути призупинені і відновлені за допомогою функцій:

SuspendThread(), **ResumeThread()**.

Дочекатися закінчення виконання потоку можна за допомогою функцій:

WaitForSingleObject(), **WaitForMultipleObjects()**.

Функції бібліотеки C для роботи з потоками

Стандартна бібліотека C розрахована на однопотоківі процеси. При роботі з кількома потоками можуть бути проблеми, пов'язані із використанням глобальної пам'яті процесу різними потоками. Для роботи з потоками слід використовувати спеціальну бібліотеку Microsoft C LI BSMТ. LI B (або sw32mt. l i b для Borland C), яка підключається до проекту через його установки. Замість функції **CreateThread()** використовують спеціальну функцію створення потоку **_beginthreadex()**. Завершується потік у такому випадку функцією **_endthreadex()** або оператором **return**. Для роботи з потоками потрібно визначити константу **_MT** (**__MT__** для Borland) і додати заголовковий файл **process. h**.

Типовий заголовок для підтримки мультипотоківості засобами бібліотеки C має вигляд:

```
#define _MT           // #define __MT__ для Borland
#include <windows.h>
#include <process.h>
```

Завдання для виконання

1. Створити програму, яка запускає кілька потоків і чекає на їх завершення. Функції потоку у вигляді параметра передати номер потоку. При успішному створенні кожного потоку вивести повідомлення. Функція потоку виводить повідомлення про початок роботи потоку та його номер, створює часову затримку за допомогою циклу і виводить повідомлення про завершення потоку із заданим номером.
2. Створити програму, первинний потік якої вводить текстовий рядок з клавіатури, запускає вторинний потік і передає йому вказівник на рядок, а сам підраховує кількість слів у рядку. Вторинний потік аналізує рядок і виводить на екран окремі слова рядка. Для аналізу рядка доцільно використати функцію **strtok()**.
3. Створити програму, первинний потік якої запускає вторинний потік, а потім приймає і виконує команди з клавіатури. Список команд: S – призупинити вторинний потік, R – звільнити потік, T – завершити потік, Q – завершити програму. Вторинний потік періодично (кожну секунду) виводить на екран час свого існування або системний час.

Контрольні питання

1. Що таке потік? Які ресурси доступні потоку?
2. Чим відрізняються процес і потік?
3. Як здійснити запуск первинного потоку процесу?
4. Як здійснити запуск інших потоків процесу?
5. Що таке функція потоку? Яка її назва? Які вона має параметри?
6. Яким чином може завершитися потік?
7. Як можна визначити момент завершення заданого потоку?
8. Як дізнатися код завершення потоку?
9. Як можна призупинити і відновити виконання потоку?