# ЛАБОРАТОРНАЯ РАБОТА №4

# СИНХРОНИЗАЦИЯ ПРОЦЕССОВ И ПОТОКОВ В ОС WINDOWS. СЕМАФОРЫ

Цель работы

Изучение программного интерфейса приложений (API) операционных систем семейства Windows. Приобретение практических навыков синхронизации потоков, с использованием семафоров в различных средах программирования.

Задачи

1. Написать программу, содержащую два потока. Первый поток генерирует последовательность чисел и помещает их в кольцевой буфер из Nbuf элементов (с проверкой на свободное место в буфере с использованием механизма семафоров). Второй считывает данные из буфера и выводит их на экран. Количество элементов, тип последовательности и длина буфера выбираются в соответствии с вариантом задания (Таблица 1);
2. Определить максимальную длину буфера при различных приоритетах потоков;

Вариант задания

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № варианта | Среда программирования | Вид последовательности | Nbuf | N |
| 11 | Visual Studio | Арифм. прогресс. с разностью 3 | 9 | 170 |

Таблица 1 – Вариант задания

Ход работы

После написания программы сперва была запущена её версия, в которой отсутствует синхронизация потоков с помощью семафоров. В результате программа вывела на экран не верные числа (Рисунок 1), что связано с отсутствием в потоках проверок заполненности буфера, а также тем, что потоки ввода-вывода перекрывали друг друга.

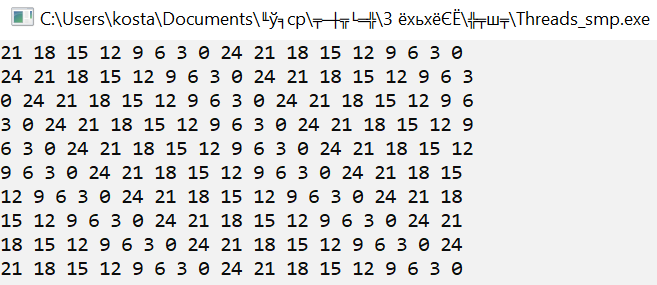


Рисунок 1 – Результат работы программы без синхронизации потоков

Затем была запущена полноценная версия программы с реализованным механизмом синхронизации потоков. В результате программа верно вывела арифметическую прогрессию, заданную вариантом (Рисунок 2).

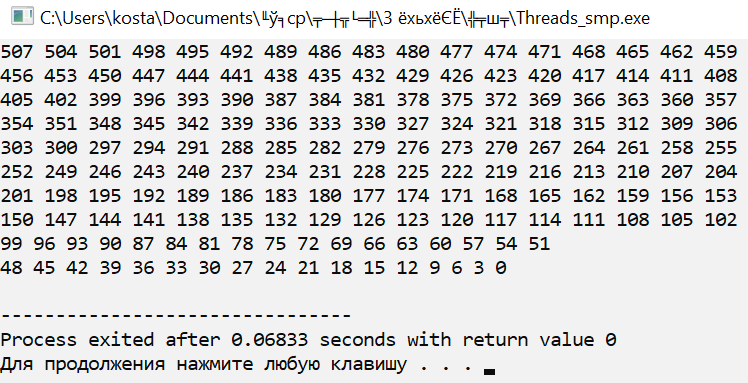


Рисунок 2 – Результат синхронизации потоков программы

Далее в программу был добавлен механизм подсчёта максимальной заполненности буфера числами (т.е. максимальное значение семафора nonf). В результате выяснилось, что вне зависимости от приоритетов потоков буфер всегда заполняется на максимум и уже затем происходит вывод его данных на экран.

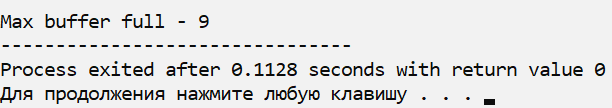


Рисунок 3 – Максимальная длина буфера

Исключением является приоритет THREAD\_PRIORITY\_TIME\_CRITICAL, при котором программа в принципе не работает, т.к. один поток не может выполниться до конца без другого.

Текст программы

#include <windows.h>

#include <iostream>

using namespace std;

// Значения, заданные по варианту

const int Nbuf = 9; // длина буфера

const int N = 170; // длина последовательности

// Буфер и его макс. заполненность

int arr[Nbuf];

int c = 0; // текущая заполненность буфера

int m = 0; // макс. заполненность буфера

// Дескрипторы семафоров

HANDLE f = CreateSemaphore(NULL, Nbuf, Nbuf, "SemaphoreFree");

HANDLE nonf = CreateSemaphore(NULL, 0, Nbuf, "SemaphoreNonFree");

HANDLE bin = CreateSemaphore(NULL, 1, 1, "SemaphoreBin");

// Поток ввода в буфер

LPTHREAD\_START\_ROUTINE input()

{

int n = 510;

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

{

WaitForSingleObject(f, INFINITE);

WaitForSingleObject(bin, INFINITE);

n -= 3;

arr[i % Nbuf] = n;

c++;

if (c > m) m = c;

ReleaseSemaphore(bin, 1, NULL);

ReleaseSemaphore(nonf, 1, NULL);

}

return 0;

}

// Поток вывода из буфера

LPTHREAD\_START\_ROUTINE output()

{

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

{

WaitForSingleObject(nonf, INFINITE);

WaitForSingleObject(bin, INFINITE);

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

if (((i + 1) % 17 == 0) && (i)) cout << endl;

c--;

ReleaseSemaphore(bin, 1, NULL);

ReleaseSemaphore(f, 1, NULL);

}

return 0;

}

// Главная функция

int main()

{

DWORD id[2]; // идентификаторы

HANDLE th[2]; // дескрипторы

// Создание потоков

th[0] = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)input, (LPVOID)(0), THREAD\_PRIORITY\_NORMAL, &id[0]);

th[1] = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)output, (LPVOID)(0), THREAD\_PRIORITY\_NORMAL, &id[1]);

WaitForMultipleObjects(2, th, TRUE, INFINITE);

// Максимальная заполненность (длина) буфера

cout << endl << "Max buffer full - " << m;

// Закрыть дескрипторы семафоров

CloseHandle(f);

CloseHandle(nonf);

CloseHandle(bin);

}

Вывод

В ходе работы были получены навыки синхронизации процессов и потоков с помощью семафоров. Полученные знания были использованы для написания программы, решающей стандартную задачу «производитель-потребитель» с набором чисел путём синхронизации их ввода и вывода из установленного буфера.