Министерство образования и науки РФ

Севастопольский государственный университет

Отчёт

по лабораторной №4

**«Синхронизация процессов и потоков в ОС Windows.**

**Семафоры»**

Выполнил:

ст.гр.ИСб-22д

Воронин И.Ю.

Проверил:

Забаштанский А.К.

Севастополь

2015

1.Цель работы

Изучение программного интерфейса приложений(API) операционных систем Windows 9x, NT, ME, XP. Приобретение практических навыков синхронизации процессов и потоков, используя Win API в средах программирования Borland Delphi, C++ Builder или Visual Studio.

2.Вариант задания

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.

\*) при заполнении кольцевого буфера добавление элементов продолжается сначала, т.е. для вычисления индекса очередного элемента используется операция деления по модулю Nbuf(Pascal: i mod Nbuf; Cи: I % Nbuf).

Количество элементов, тип последовательности и длина буфера выбираются в соответствии с вариантом задания (таблица 3.1).

Определить максимальную длину буфера при различных приоритетах оттоков.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | C++ Builder | Арифм. Прогресс. С разностью 4 | 8 | 100 |

3.Текст программы

#include <iostream>

#include <windows.h>

using namespace std;

const int N\_BUF = 8;

const int N=100;

int buf[N\_BUF];

DWORD WINAPI TH1(const LPVOID param){

HANDLE sem = OpenSemaphore(SYNCHRONIZE,FALSE,"sem");

int n=0,i=0,k=0;

while(i<=N){

WaitForSingleObject(sem,INFINITE);

if(k==8){

k=0;

ReleaseSemaphore(sem,1,NULL);

Sleep(100);

continue;

}

buf[i%N\_BUF]= n;

n += 4;

i++;

k++;

}

}

DWORD WINAPI TH2(const LPVOID param){

HANDLE sem = OpenSemaphore(SYNCHRONIZE,FALSE,"sem");

int i=-1;

int k=0;

while(i<N){

if(k==8){

k=0;

cout << endl;

ReleaseSemaphore(sem,1,NULL);

Sleep(100);

continue;

}

k++;

WaitForSingleObject(sem,INFINITE);

cout << buf[++i%N\_BUF] << ' ';

}

}

int main(){

HANDLE sem = CreateSemaphore(NULL,1,1,"sem");

HANDLE T1 = CreateThread(NULL,0,&TH1,0,CREATE\_SUSPENDED,0);

HANDLE T2 = CreateThread(NULL,0,&TH2,0,CREATE\_SUSPENDED,0);

SetThreadPriority(T1,THREAD\_PRIORITY\_HIGHEST);

SetThreadPriority(T2,THREAD\_PRIORITY\_LOWEST);

ResumeThread(T1);

ResumeThread(T2);

CloseHandle(sem);

cin.get();

return 0;

}

4.Тест программы

Тест программы c использованием семафоры(рис.4.1)

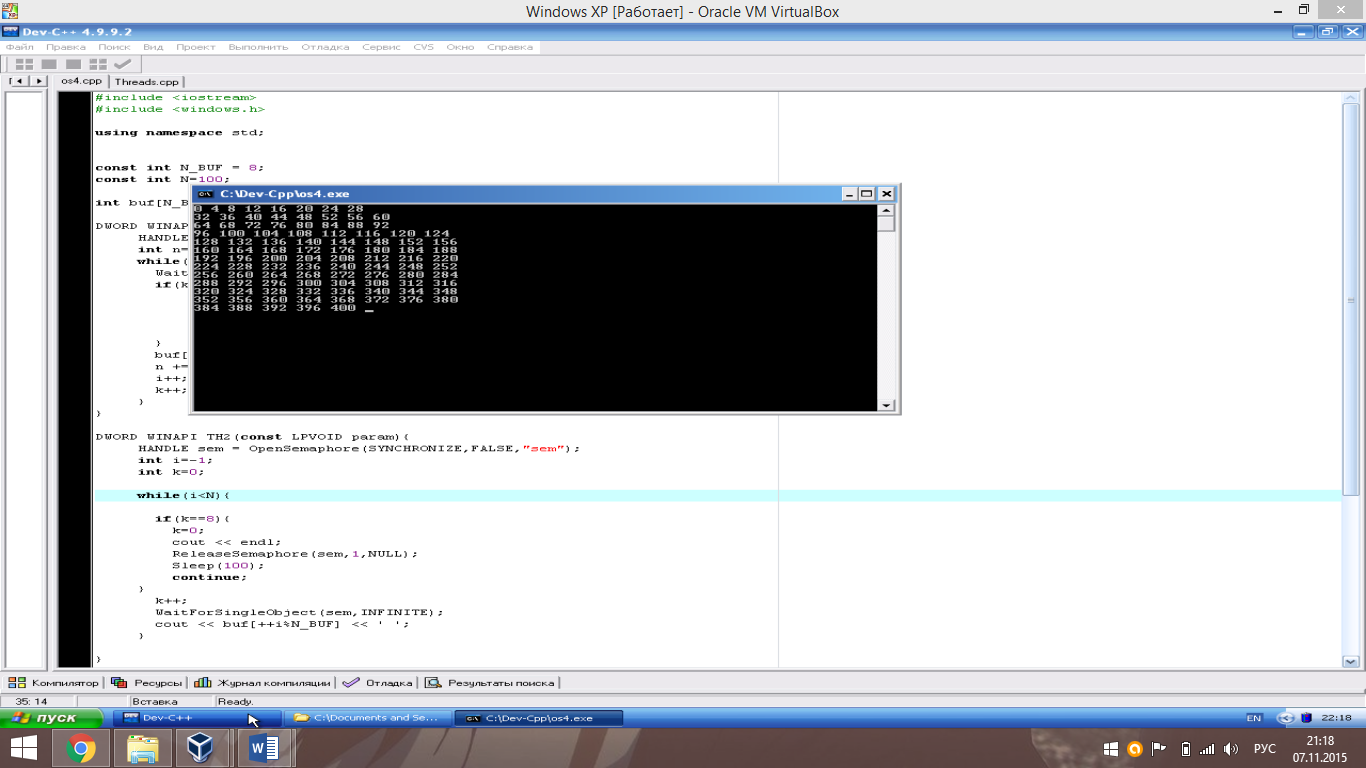


Рис.4.1-Тест.

В данном случае основная программа запускает два потока, которые поочерёдно обращаются к буферу. Сначала первый поток записывает туда данные, а потом второй выводит их на экран. Данные действия согласовываются при помощи семафоры.

ВЫВОДЫ

В данной лабораторной работе были изучены понятия семафор и применены на практике при использовании двух потоков с общим ресурсом. За счёт семафор данные потоки были согласованы. Программа была написана на языке C++.