НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №2

з дисципліни **«**Програмування паралельних комп’ютерних систем**»**

**Бібліотека Win32.**

Виконав:

студент 3 курсу

ФІОТ гр. ІО-34

Кривоносов Олексій

Перевірив:

Корочкін О. В.

Київ – 2016 р.

**Тема:** Програмування для комп’ютерних систем зі спільною пам’яттю. Бібліотека Win32

**Технічне завдання:** Розробити програму для розв’язання ПКС із СП математичної задачі: A=(B\*C)\*Z+α\*E\*(MX\*MZ)

Бібліотека: Win32.

Засоби організації взаємодії: семафори, мютекси, критичні секції, події

Засоби взаємодії: семафори.



Структурна схема ПКС

**Виконання роботи:**

**Етап 1. Побудова паралельного алгоритму**

1. bi = Bн\*Cн  (i = 1..P)
2. b= b+bi (i = 1..P)
3. AН = b\*ZH + α\*E\*(MX\*MZH)

Спільний ресурс: MX, b, E, α

**Етап 2. Розроблення алгоритмів роботи кожного процесу**

|  |  |
| --- | --- |
| **Задача Т1** | ТС, КД |
| 1. Введення B, MZ |  |
| 2. Сигнал задачам Т2, Т3, Т4 про введення B, MZ | S2,3,4 – 1 |
| 3. Чекати на введення C, Z у задачі Т3 | W3 – 1 |
| 4. Чекати на введення α, E, MX у задачі T4 | W4 – 2 |
| 5. Обчислення: b1 = Bн\*Cн |  |
| 6. Обчислення: b = b + b1 | КД |
| 7. Сигнал задачам Т2, Т3, Т4 про обчислення b | S2,3,4 – 1 |
| 8. Чекати на завершення обчислень b в Т2 | W2– 3 |
| 9. Чекати на завершення обчислень b в Т3 | W3 – 4 |
| 10. Чекати на завершення обчислень b в Т4 | W4 – 5 |
| 11. Копіювання: E1=E, α1= α, b1 := b | КД |
| 12. Копіювання: МХ1 := МХ | КД |
| 13. Обчислення: AН = b1\*ZH + α1\*E1\*(MX1\*MZH) |  |
| 14. Чекати на завершення обчислень A в T2, T3, T4 | W2,3,4 – 6 |
| 15. Виведення A |  |
| **Задача T2** | ТС, КД |
| 1. Чекати на введення B, MZ у задачі T1 | W1 – 1 |
| 2. Чекати на введення C, Z у задачі T3 | W3 – 2 |
| 3. Чекати на введення α, E, MX у задачі T4 | W4 – 3 |
| 4. Обчислення: b2 = Bн\*Cн |  |
| 5. Обчислення: b = b + b2 | КД |
| 6. Сигнал задачам Т1, Т3, Т4 про обчислення b | S1,3,4 – 1 |
| 7. Чекати на завершення обчислень b в Т1 | W1– 4 |
| 8. Чекати на завершення обчислень b в Т3 | W3– 5 |
| 9. Чекати на завершення обчислень b в Т4 | W4– 6 |
| 10. Копіювання: E2=E, α2= α, b2 = b | КД |
| 11. Копіювання: МХ2 := МХ | КД |
| 12. Обчислення: AН = b2\*ZH + α2\*E2\*(MX2\*MZH) |  |
| 13. Сигнал T1 про завершення обчислень A | S1 – 2 |
| **Задача Т3** | ТС, КД |
| 1. Введення C, Z |  |
| 2. Сигнал задачам Т1, Т2, Т4 про введення C, Z | S1,2,4 – 1 |
| 3. Чекати на введення B, MZ у задачі T1 | W1 – 1 |
| 4. Чекати на введення α, E, MX у задачі T4 | W4 – 2 |
| 5. Обчислення: b3 = Bн\*Cн |  |
| 6. Обчислення: b = b + b3 | КД |
| 7. Сигнал задачам Т1, Т2, Т4 про обчислення b | S1,2,4 – 2 |
| 8.Чекати на завершення обчислень b в Т1 | W1 – 3 |
| 9.Чекати на завершення обчислень b в Т2 | W2– 4 |
| 10. Чекати на завершення обчислень b в Т4 | W4– 5 |
| 11. Копіювання:E3=E, α3= α, b3 = b | КД |
| 12. Копіювання: МХ3 := МХ | КД |
| 13. Обчислення: AН = b3\*Z3 + α3\*E3\*(MX3\*MZH) |  |
| 14. Сигнал T1 про завершення обчислень A | S1 – 3 |
| **Задача Т4** | ТС, КД |
| 1. Введення α, E, MX |  |
| 2. Сигнал задачам T1, T2, T3 про введення α, E, MX | S1,2,3 – 1 |
| 3. Чекати на введення B, MZ у задачі T1 | W1 – 1 |
| 4. Чекати на введення C, Z у задачі T3 | W3 – 2 |
| 5. Обчислення: b4 = Bн\*Cн |  |
| 6. Обчислення: b = b + b4 | КД |
| 7. Сигнал задачам Т1, Т2, Т3 про обчислення b | S1,2,3 – 2 |
| 8. Чекати на завершення обчислень b в Т1 | W1– 3 |
| 9. Чекати на завершення обчислень b в Т2 | W2– 4 |
| 10. Чекати на завершення обчислень b в Т3 | W3– 5 |
| 11. Копіювання: E4=E, α4= α, b4=b | КД |
| 12. Копіювання: МХ4 := МХ | КД |
| 13. Обчислення: AН = b4\*ZH+ α4\*E4\*(MX4\*MZH) |  |
| 14. Сигнал T1 про завершення обчислень A | S1 – 3 |

**Етап 3. Розроблення структурної схеми взаємодії задач**

На структурній схемі взаємодії задач уведено такі умовні позначення

* Evn1\_234 – подія для синхронізації із завершенням вводу в T1
* Evn3\_124 – подія для синхронізації із завершенням вводу в T3
* Evn4\_123 – подія для синхронізації із завершенням вводу в T4
* CrSec – для доступу до спільного ресурсу E, α, b
* Mute – для доступу до спільного ресурсу MX
* EvnB1\_234 – подія для синхронізації із завершенням обчислення b1 в Т1
* EvnB2\_134 – подія для синхронізації із завершенням обчислення b1 в Т2
* EvnB3\_124 – подія для синхронізації із завершенням обчислення b1 в Т3
* EvnB4\_123 – подія для синхронізації із завершенням обчислення b1 в Т4
* Sem\_A[3] – для синхронізації обчислень і виведення результату

Етап 4. Розроблення програми

/\*------------------------------------------------------------------

-- --

-- Parallel and Distributed Computing --

-- Laboratory work #2. Win32 --

-- --

-- --

-- Task: A=(B\*C)\*Z+α\*E\*(MX\*MZ) --

-- --

-- Author: Krivonosov Olexiy, group IO-34 --

-- Date: 5.03.2016 --

-- --

------------------------------------------------------------------ \*/

#include "stdafx.h"

#include <iostream>

#include <windows.h>

using namespace std;

typedef int\* vector;

typedef int\*\* matrix;

const int N = 4;

const int P = 4;

const int H = N / P;

int alfa, b=0;

vector B = new int[N],

C = new int[N],

E = new int[N],

A = new int[N],

Z = new int[N];

matrix MX = new vector[N],

MZ = new vector[N];

HANDLE Evn1\_234, Evn3\_124, Evn4\_123, EvnB1\_234,

EvnB2\_134, EvnB3\_124, EvnB4\_123, Mute, Sem\_A[3];

CRITICAL\_SECTION CrSec;

//-----------------------------------------T1--------------------------------------------

void T1(){

int alfa1, s, b1 = 0;

vector

E1 = new int[N],

BufV1 = new int[N];

matrix BufM1 = new vector[N];

matrix MX1 = new vector[N];

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

{

BufM1[i] = new int[N];

};

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

{

MX1[i] = new int[N];

};

cout << "Process T1 started" << endl;

//1. Введення MZ, B

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

{

MZ[i] = new int[N];

};

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

{

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

{

MZ[i][j] = 1;

}

}

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

B[i] = 1;

}

//2. Сигнал задачам Т2, Т3, Т4 про введення MZ, B

SetEvent(Evn1\_234);

//3. Чекати на введення C, Z у задачі Т3

WaitForSingleObject(Evn3\_124, INFINITE);

//4. Чекати на введення alfa, E, MX задачі T4

WaitForSingleObject(Evn4\_123, INFINITE);

//5.Обчислення: b1 = Bн\*Cн

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

b1 = b1 + B[i] \* C[i];

}

//6.Обчислення: b = b + b1

EnterCriticalSection(&CrSec);

b = b + b1;

LeaveCriticalSection(&CrSec);

//7. Сигнал задачам Т2, Т3, Т4 про обчислення b

SetEvent(EvnB1\_234);

//8. Чекати на завершення обчислень b в Т2

WaitForSingleObject(EvnB2\_134, INFINITE);

//9.Чекати на завершення обчислень b в Т3

WaitForSingleObject(EvnB3\_124, INFINITE);

//10. Чекати на завершення обчислень b в Т4

WaitForSingleObject(EvnB4\_123, INFINITE);

//11. Копіювання: E1 = E, α1 = α, b1 : = b

EnterCriticalSection(&CrSec);

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

E1[i] = E[i];

}

alfa1 = alfa;

b1 = b;

LeaveCriticalSection(&CrSec);

//12. Копіювання: МХ1 : = МХ

WaitForSingleObject(Mute, INFINITE);

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

for (int j = 0; j < N; j++){

MX1[i] = MX[i];

}

}

ReleaseMutex(Mute);

//13. Обчислення: AН = b1\*ZH + α1\*E1\*(MX1\*MZH)

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

{

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

{

s = 0;

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

{

s += MZ[i][k] \* MX1[k][j];

}

BufM1[i][j] = s;

}

}

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

s = 0;

for (int j = 0; j < N; j++){

s += BufM1[i][j] \* E[j];

}

BufV1[i] = s\*alfa1;

}

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

Z[i] = Z[i] \* b1;

}

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

A[i] = Z[i] + BufV1[i];

}

//14.Чекати на завершення обчислень A в T2, T3, T4

WaitForMultipleObjects(3, Sem\_A, TRUE, INFINITE);

//15. Виведення A

if (N<10)

{

cout << " " << endl;

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

{

cout << A[i];

cout << " ";

}

cout << " " << endl;

}

cout << "Process T1 finished" << endl;

}

//-------------------------------------------T2-------------------------------------------

void T2(){

int alfa2, s,b2=0;

vector

E2 = new int[N],

BufV2 = new int[N],

Buf2 = new int[N],

Buf3 = new int[N];;

matrix BufM2 = new vector[N];

matrix MX2 = new vector[N];

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

{

BufM2[i] = new int[N];

};

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

{

MX2[i] = new int[N];

};

cout << "Process T2 started" << endl;

//1. Чекати на введення B, MZ у задачі T1

WaitForSingleObject(Evn1\_234, INFINITE);

//2. Чекати на введення C, Z у задачі T3

WaitForSingleObject(Evn3\_124, INFINITE);

//3. Чекати на введення α, E, MX у задачі T4

WaitForSingleObject(Evn4\_123, INFINITE);

//4. Обчислення: b2 = Bн\*Cн

for (int i = H; i < 2\*H; i++) {

b2 = b2 + B[i] \* C[i];

}

//5. Обчислення: b = b + b2

EnterCriticalSection(&CrSec);

b = b + b2;

LeaveCriticalSection(&CrSec);

//6. Сигнал задачам Т1, Т3, Т4 про обчислення b

SetEvent(EvnB2\_134);

//7.Чекати на завершення обчислень b в Т1

WaitForSingleObject(EvnB1\_234, INFINITE);

//8.Чекати на завершення обчислень b в Т3

WaitForSingleObject(EvnB3\_124, INFINITE);

//9. Чекати на завершення обчислень b в Т4

WaitForSingleObject(EvnB4\_123, INFINITE);

//10. Копіювання: E2=E, α2= α, b2 = b

EnterCriticalSection(&CrSec);

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

E2[i] = E[i];

}

alfa2 = alfa;

b2 = b;

LeaveCriticalSection(&CrSec);

//11. Копіювання: МХ2 = МХ

WaitForSingleObject(Mute, INFINITE);

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

for (int j = 0; j < N; j++){

MX2[i] = MX[i];

}

}

ReleaseMutex(Mute);

//12. Обчислення: AН = b2\*ZH + α2\*E2\*(MX2\*MZH)

for (int i = H; i < H\*2; i++)

{

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

{

s = 0;

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

{

s += MZ[i][k] \* MX2[k][j];

}

BufM2[i][j] = s;

}

}

for (int i = H; i < 2\*H; i++){

s = 0;

for (int j = 0; j < N; j++){

s += BufM2[i][j] \* E[j];

}

BufV2[i] = s\*alfa2;

}

for (int i = H; i < 2\*H; i++){

Z[i] = Z[i] \* b2;

}

for (int i = H; i < 2\*H; i++){

A[i] = Z[i] + BufV2[i];

}

//13. Сигнал T1 про завершення обчислень MA

ReleaseSemaphore(Sem\_A[0], 1, NULL);

cout << "Process T2 finished" << endl;

}

//----------------------------------------T3----------------------------------------------

void T3(){

int alfa3, s,b3 = 0;

vector

E3 = new int[N],

BufV3 = new int[N];

matrix BufM3 = new vector[N];

matrix MX3 = new vector[N];

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

{

BufM3[i] = new int[N];

};

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

{

MX3[i] = new int[N];

};

cout << "Process T3 started" << endl;

//1. Введення C, Z

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

{

C[i] = 1;

Z[i] = 1;

}

//2. Сигнал задачам Т1, Т3, Т4 про введення C, Z

SetEvent(Evn3\_124);

//3. Чекати на введення B, MZ у задачі T1

WaitForSingleObject(Evn1\_234, INFINITE);

//4. Чекати на введення α, E, MX задачі T4

WaitForSingleObject(Evn4\_123, INFINITE);

//5. Обчислення: b3 = Bн\*Cн

for (int i = H\*2; i < H\*3; i++) {

b3 = b3 + B[i] \* C[i];

}

//6. Обчислення: b = b + b3

EnterCriticalSection(&CrSec);

b = b + b3;

LeaveCriticalSection(&CrSec);

//7. Сигнал задачам Т1, Т2, Т4 про обчислення b

SetEvent(EvnB3\_124);

//8.Чекати на завершення обчислень b в Т1

WaitForSingleObject(EvnB1\_234, INFINITE);

//9.Чекати на завершення обчислень b в Т2

WaitForSingleObject(EvnB2\_134, INFINITE);

//10. Чекати на завершення обчислень b в Т4

WaitForSingleObject(EvnB4\_123, INFINITE);

//11. Копіювання: E3=E, α3= α, b3 = b

EnterCriticalSection(&CrSec);

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

E3[i] = E[i];

}

alfa3 = alfa;

b3 = b;

LeaveCriticalSection(&CrSec);

//12. Копіювання: МХ3 : = МХ

WaitForSingleObject(Mute, INFINITE);

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

for (int j = 0; j < N; j++){

MX3[i] = MX[i];

}

}

ReleaseMutex(Mute);

//13. Обчислення: AН = b3\*Z3 + α3\*E3\*(MX3\*MZH)

for (int i = H\*2; i < H\*3; i++)

{

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

{

s = 0;

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

{

s += MZ[i][k] \* MX3[k][j];

}

BufM3[i][j] = s;

}

}

for (int i = 2\*H; i < H\*3; i++){

s = 0;

for (int j = 0; j < N; j++){

s += BufM3[i][j] \* E[j];

}

BufV3[i] = s\*alfa3;

}

for (int i = H\*2; i < H\*3; i++){

Z[i] = Z[i] \* b3;

}

for (int i = H\*2; i < H\*3; i++){

A[i] = Z[i] + BufV3[i];

}

//14. Сигнал T1 про завершення обчислень MA

ReleaseSemaphore(Sem\_A[1], 1, NULL);

cout << "Process T3 finished" << endl;

}

//---------------------------------------T4-------------------------------------------------

void T4(){

int alfa4, s,b4=0;

vector B4 = new int[N],

E4 = new int[N],

C4 = new int[N],

BufV4 = new int[N];

matrix BufM4 = new vector[N];

matrix MX4 = new vector[N];

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

{

BufM4[i] = new int[N];

};

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

{

MX4[i] = new int[N];

};

cout << "Process T4 started" << endl;

//1. Введення α, E, MX

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

{

MX[i] = new int[N];

};

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

{

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

{

MX[i][j] = 1;

}

}

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

E[i] = 1;

}

alfa = 1;

//2. Сигнал задачам T1, T2, T3 про введення α, E, MX

SetEvent(Evn4\_123);

//3. Чекати на введення B, MZ у задачі T1

WaitForSingleObject(Evn1\_234, INFINITE);

//4. Чекати на введення C, Z у задачі T3

WaitForSingleObject(Evn3\_124, INFINITE);

//5. Обчислення: b4 = Bн\*Cн

for (int i = H \*3; i < N; i++) {

b4 = b4 + B[i] \* C[i];

}

//6. Обчислення: b = b + b4

EnterCriticalSection(&CrSec);

b = b + b4;

LeaveCriticalSection(&CrSec);

//7. Сигнал задачам Т1, Т2, Т3 про обчислення b

SetEvent(EvnB4\_123);

//8.Чекати на завершення обчислень b в Т1

WaitForSingleObject(EvnB1\_234, INFINITE);

//9.Чекати на завершення обчислень b в Т2

WaitForSingleObject(EvnB2\_134, INFINITE);

//10. Чекати на завершення обчислень b в Т3

WaitForSingleObject(EvnB3\_124, INFINITE);

//11. Копіювання: E4 = E, α4 = α, b4 = b

EnterCriticalSection(&CrSec);

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

E4[i] = E[i];

}

alfa4 = alfa;

b4 = b;

LeaveCriticalSection(&CrSec);

//12. Копіювання: МХ4 : = МХ

WaitForSingleObject(Mute, INFINITE);

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

for (int j = 0; j < N; j++){

MX4[i] = MX[i];

}

}

ReleaseMutex(Mute);

//13. Обчислення: AН = b4\*ZH+ α4\*E4\*(MX4\*MZH)

for (int i = H\*3; i < N; i++)

{

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

{

s = 0;

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

{

s += MZ[i][k] \* MX4[k][j];

}

BufM4[i][j] = s;

}

}

for (int i = 3 \* H; i < N; i++){

s = 0;

for (int j = 0; j < N; j++){

s += BufM4[i][j] \* E[j];

}

BufV4[i] = s\*alfa4;

}

for (int i = 3\*H; i < N; i++){

Z[i] = Z[i] \* b4;

}

for (int i = 3\*H; i < N; i++){

A[i] = Z[i] + BufV4[i];

}

//14. Сигнал T1 про завершення обчислень MA

ReleaseSemaphore(Sem\_A[2], 1, NULL);

cout << "Process T4 finished" << endl;

}

int main(int argc, char\* argv[])

{

cout << "Lab 2 started" << endl;

Evn4\_123 = CreateEvent(NULL, TRUE, FALSE, NULL);

Evn3\_124 = CreateEvent(NULL, TRUE, FALSE, NULL);

Evn1\_234 = CreateEvent(NULL, TRUE, FALSE, NULL);

EvnB1\_234 = CreateEvent(NULL, TRUE, FALSE, NULL);

EvnB2\_134 = CreateEvent(NULL, TRUE, FALSE, NULL);

EvnB3\_124 = CreateEvent(NULL, TRUE, FALSE, NULL);

EvnB4\_123 = CreateEvent(NULL, TRUE, FALSE, NULL);

InitializeCriticalSection(&CrSec);

Mute = CreateMutex(NULL, FALSE, NULL);

Sem\_A[0] = CreateSemaphore(NULL, 0, 1, NULL);

Sem\_A[1] = CreateSemaphore(NULL, 0, 1, NULL);

Sem\_A[2] = CreateSemaphore(NULL, 0, 1, NULL);

DWORD Tid1, Tid2, Tid3, Tid4;

HANDLE threads[] =

{

CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)T1, NULL, NULL, &Tid1),

CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)T2, NULL, NULL, &Tid2),

CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)T3, NULL, NULL, &Tid3),

CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)T4, NULL, NULL, &Tid4)

};

WaitForMultipleObjects(4, threads, true, INFINITE);

CloseHandle(threads[0]);

CloseHandle(threads[1]);

CloseHandle(threads[2]);

CloseHandle(threads[3]);

cout << "Lab2 ended" << endl;

char key;

cin >> key;

return 0;

}