НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №2

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

Виконав:

студент 3 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Перевірив:

Корочкін О. В.

Київ – 2015 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: MA = MB∙MC∙α + β∙max(Z)∙MK.

Бібліотека: Win32.

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

Засоби взаємодії: семафори.



Структурна схема ПКС

**Виконання роботи:**

**Етап 1. Побудова паралельного алгоритму**

1. mi = max(ZH), i =
2. m = max(mi, m)
3. MAH = MBH∙MC∙α + β∙m∙MKH

Спільний ресурс: m, α, β, MC

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

|  |  |
| --- | --- |
| **Задача Т1** | ТС, КД |
| 1. Введення MB |  |
| 2. Сигнал задачам Т2, Т3, Т4 про введення MB | S2,3,4 – 1 |
| 3. Чекати на введення MK, MC, α у задачі Т3 | W3 – 1 |
| 4. Чекати на введення Z, β у задачі T4 | W4 – 2 |
| 5. Копіювати MC1 := MC | КД |
| 6. Копіювати α1:= α | КД |
| 7. Копіювати β1:= β | КД |
| 8. Обчислення m1 := max(ZH) |  |
| 9. Обчислення m: = max (m, m1) | КД |
| 10. Чекати на завершення обчислень m в T2, T3, T4 | W2,3,4 – 3 |
| 11. Копіювання m1: = m | КД |
| 12. Сигнал T2, T3, T4 про завершення обчислень m | S2,3,4 – 2 |
| 13. Обчислення MAH = MBH∙MC1∙α1 + β1∙m1∙MKH |  |
| 14. Чекати на завершення обчислень MA в T2, T3, T4 | W2,3,4 – 4 |
| 15. Виведення MA |  |
| **Задача T2** | ТС, КД |
| 1. Чекати на введення MB у задачі T1 | W1 – 1 |
| 2. Чекати на введення MK, MC, α у задачі T3 | W3 – 2 |
| 3. Чекати на введення Z, β у задачі T4 | W4 – 3 |
| 4. Копіювати MC2:= MC |  |
| 5. Копіювати α2:= α | КД |
| 6. Копіювати β2:= β | КД |
| 7. Обчислення m2 := max(ZH) |  |
| 8. Обчислення m: = max(m,m2) | КД |
| 9. Сигнал Т1 про про завершення обчислень m | S1 – 2 |
| 10. Чекати сигналу від T1 про завершення обчислень m | W1 – 4 |
| 11. Копіювання m2: = m | КД |
| 12. Обчислення MAH = MBH∙MC2∙α2 + β2∙m2∙MKH |  |
| 13. Сигнал T1 про завершення обчислень MA | S1 – 3 |
| **Задача Т3** | ТС, КД |
| 1. Введення МК, МС, α |  |
| 2. Сигнал задачам Т1, Т3, Т4 про введення МК, МС, α | S1,3,4 – 1 |
| 3. Чекати на введення MB у задачі T1 | W1 – 1 |
| 4. Чекати на введення Z, β у задачі T4 | W4 – 2 |
| 5. Копіювати MC3:= MC | КД |
| 6. Копіювати α3:= α | КД |
| 7. Копіювати β3:= β | КД |
| 8. Обчислення m3 := max(ZH) |  |
| 9. Обчислення m: = max(m,m3) | КД |
| 10. Сигнал Т1 про про завершення обчислень m | S1 – 2 |
| 11. Чекати сигналу від T1 про завершення обчислень m | W1 – 3 |
| 12. Копіювання m3: = m | КД |
| 13. Обчислення MAH = MBH∙MC3∙α3 + β3∙m3∙MKH |  |
| 14. Сигнал T1 про завершення обчислень MA | S1 – 3 |
| **Задача Т4** | ТС, КД |
| 1. Введення Z, β |  |
| 2. Сигнал задачам T1, T2, T3 про введення Z, β | S1,2,3 – 1 |
| 3. Чекати на введення MB у задачі T1 | W1 – 1 |
| 4. Чекати на введення MK, MC, α у задачі T3 | W3 – 2 |
| 5. Копіювати MC4:= MC | КД |
| 6. Копіювати α4:= α | КД |
| 7. Копіювати β4:= β | КД |
| 8. Обчислення m4 := max(ZH) |  |
| 9. Обчислення m: = max(m,m4) | КД |
| 10. Сигнал Т1 про про завершення обчислень m | S1 – 2 |
| 11. Чекати сигналу від T1 про завершення обчислень m | W1 – 3 |
| 12. Копіювання m4: = m | КД |
| 13. Обчислення MAH = MBH∙MC4∙α4 + β4∙m4∙MKH |  |
| 14. Сигнал T1 про завершення обчислень MA | 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 α, β, m
* Mute – для доступу до спільного ресурсу MC
* Sem\_m[3], Sem\_max – для синхронізації обчислень максимуму вектора Z
* Sem\_MA[3] – для синхронізації решти обчислень і виведення результату



Структурна схема взаємодії задач

Етап 4. Розроблення програми

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

-- --

-- Parallel and Distributed Computing --

-- Laboratory work #2. Win32 --

-- --

-- --

-- Task: MA = MB\*MC\*a+bmax(Z)\*MK --

-- --

-- Author: Kuzmenko Volodimir, group IO-21 --

-- Date: 11.03.2015 --

-- --

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

#include "stdafx.h"

#include <iostream>

#include <windows.h>

using namespace std;

typedef int\* vector;

typedef int\*\* matrix;

const int N = 4000;

const int P = 4;

const int H = N/P;

int alfa, beta, m;

vector Z = new int [N];

matrix MA = new vector[N],

MB = new vector[N],

MC = new vector [N],

MK = new vector [N];

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

Mute, Sem\_m[3], Sem\_MA[3],Sem\_max;

CRITICAL\_SECTION CrSec;

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

void T1 (){

int alfa1, beta1, m1, s;

matrix MC1 = new vector [N];

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

{

MC1[i] = new int[N];

};

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

//1. Введення MB

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

{

MB[i] = new int [N];

};

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

{

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

{

MB[i][j] = 1;

}

}

//2. Сигнал задачам Т2, Т3, Т4 про введення MB

SetEvent(Evn1\_234);

//3. Чекати на введення MK, MC, α у задачі Т3

WaitForSingleObject(Evn3\_124, INFINITE);

//4. Чекати на введення Z, β у задачі T4

WaitForSingleObject(Evn4\_123, INFINITE);

//5. Копіювати MC1 := MC

WaitForSingleObject(Mute, INFINITE);

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

{

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

{

MC1[i][j] = MC[i][j];

}

}

ReleaseMutex(Mute);

//6, 7. Копіювання alfa, beta

EnterCriticalSection(&CrSec);

alfa1 = alfa;

beta1 = beta;

LeaveCriticalSection(&CrSec);

//8. Обчислення m1 := max(ZH)

m1=0;

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

{

if (Z[i]>m1){

m1 = Z[i];

}

}

//9. Обчислення m: = max (m, m1)

EnterCriticalSection(&CrSec);

m = max(m, m1);

LeaveCriticalSection(&CrSec);

//10. Чекати на завершення обчислень m в T2, T3, T4

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

//11. Копіювання m1: = m

EnterCriticalSection(&CrSec);

m1 = m;

LeaveCriticalSection(&CrSec);

//12. Сигнал T2, T3, T4 про завершення обчислень m

ReleaseSemaphore(Sem\_max, 1, NULL);

//13. Обчислення MAH = MBH∙MC1∙α1 + β1∙m1∙MKH

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

{

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

{

s=0;

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

{

s+=MB[i][k]\*MC1[k][j];

}

MA[i][j]=s\*alfa1+beta1\*m1\*MB[i][j];

}

}

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

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

//15. Виведення MA

if (N<10)

{

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

{

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

{

cout << MA[i][j];

}

cout <<endl;

}

}

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

}

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

void T2 (){

int alfa2, beta2, s, m2;

matrix MC2 = new vector [N];

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

{

MC2[i] = new int[N];

};

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

//1. Чекати на введення MB у задачі T1

WaitForSingleObject(Evn1\_234, INFINITE);

//2. Чекати на введення MK, MC, α у задачі T3

WaitForSingleObject(Evn3\_124, INFINITE);

//3. Чекати на введення Z, β у задачі T4

WaitForSingleObject(Evn4\_123, INFINITE);

//4. Копіювати MC2:= MC

WaitForSingleObject(Mute, INFINITE);

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

{

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

{

MC2[i][j] = MC[i][j];

}

}

ReleaseMutex(Mute);

//5,6. Копіювання alfa, beta

EnterCriticalSection(&CrSec);

alfa2 = alfa;

beta2 = beta;

LeaveCriticalSection(&CrSec);

//7. Обчислення m2 := max(ZH)

m2=0;

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

{

if (Z[i]>m2){

m2 = Z[i];

}

}

//8. Обчислення m: = max(m,m2)

EnterCriticalSection(&CrSec);

m = max(m, m2);

LeaveCriticalSection(&CrSec);

//9. Сигнал Т1 про про завершення обчислень m

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

//10. Чекати сигналу від T1 про завершення обчислень m

WaitForSingleObject(Sem\_max, INFINITE);

//11. Копіювання m2: = m

EnterCriticalSection(&CrSec);

m2 = m;

LeaveCriticalSection(&CrSec);

//12. Обчислення MAH = MBH∙MC2∙α2 + β2∙m2∙MKH

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

{

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

{

s=0;

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

{

s+=MB[i][k]\*MC2[k][j];

}

MA[i][j]=s\*alfa2+beta2\*m2\*MB[i][j];

}

}

//13. Сигнал T1 про завершення обчислень MA

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

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

}

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

void T3(){

int alfa3, beta3, s, m3;

matrix MC3 = new vector [N];

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

{

MC3[i] = new int[N];

MA[i] = new int [N];

};

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

//1. Введення МК, МС, α

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

{

MB[i] = new int[N];

MC[i] = new int[N];

};

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

{

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

{

MB[i][j] = 1;

MC[i][j] = 1;

}

}

alfa=1;

//2. Сигнал задачам Т1, Т3, Т4 про введення МК, МС, α

SetEvent(Evn3\_124);

//3. Чекати на введення MB у задачі T1

WaitForSingleObject(Evn3\_124, INFINITE);

//4. Чекати на введення Z, β у задачі T4

WaitForSingleObject(Evn4\_123, INFINITE);

//5. Копіювати MC3:= MC

WaitForSingleObject(Mute, INFINITE);

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

{

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

{

MC3[i][j] = MC[i][j];

}

}

ReleaseMutex(Mute);

//6,7. Копіювання alfa, beta

EnterCriticalSection(&CrSec);

alfa3 = alfa;

beta3 = beta;

LeaveCriticalSection(&CrSec);

//8. Обчислення m3 := max(ZH)

m3=0;

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

{

if (Z[i]>m3){

m3 = Z[i];

}

}

//9. Обчислення m: = max(m,m3)

EnterCriticalSection(&CrSec);

m = max(m, m3);

LeaveCriticalSection(&CrSec);

//10. Сигнал Т1 про про завершення обчислень m

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

//11. Чекати сигналу від T1 про завершення обчислень m

WaitForSingleObject(Sem\_max, INFINITE);

//12. Копіювання m3: = m

EnterCriticalSection(&CrSec);

m3 = m;

LeaveCriticalSection(&CrSec);

//13. Обчислення MAH = MBH∙MC3∙α3 + β3∙m3∙MKH

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

{

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

{

s=0;

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

{

s+=MB[i][k]\*MC3[k][j];

}

MA[i][j]=s\*alfa3+beta3\*m3\*MB[i][j];

}

}

//14. Сигнал T1 про завершення обчислень MA

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

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

}

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

void T4(){

int alfa4, beta4, s, m4;

matrix MC4 = new vector [N];

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

{

MC4[i] = new int[N];

};

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

//1. Введення Z, β

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

{

Z[i] = 1;

}

beta=1;

//2. Сигнал задачам T1, T2, T3 про введення Z, β

SetEvent(Evn4\_123);

//3. Чекати на введення MB у задачі T1

WaitForSingleObject(Evn1\_234, INFINITE);

//4. Чекати на введення MK, MC, α у задачі T3

WaitForSingleObject(Evn3\_124, INFINITE);

//5. Копіювати MC4:= MC

WaitForSingleObject(Mute, INFINITE);

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

{

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

{

MC4[i][j] = MC[i][j];

}

}

ReleaseMutex(Mute);

//5, 6. Копіювання alfa, beta

EnterCriticalSection(&CrSec);

alfa4 = alfa;

beta4 = beta;

LeaveCriticalSection(&CrSec);

//8. Обчислення m4 := max(ZH)

m4=0;

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

{

if (Z[i]>m4){

m4 = Z[i];

}

}

//9. Обчислення m: = max(m,m4)

EnterCriticalSection(&CrSec);

m = max(m, m4);

LeaveCriticalSection(&CrSec);

//10. Сигнал Т1 про про завершення обчислень m

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

//11. Чекати сигналу від T1 про завершення обчислень m

WaitForSingleObject(Sem\_max, INFINITE);

//12. Копіювання m4: = m

EnterCriticalSection(&CrSec);

m4 = m;

LeaveCriticalSection(&CrSec);

//13. Обчислення MAH = MBH∙MC4∙α4 + β4∙m4∙MKH

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

{

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

{

s=0;

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

{

s+=MB[i][k]\*MC4[k][j];

}

MA[i][j]=s\*alfa4+beta4\*m4\*MB[i][j];

}

}

//14. Сигнал T1 про завершення обчислень MA

ReleaseSemaphore(Sem\_MA[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);

InitializeCriticalSection(&CrSec);

Mute = CreateMutex(NULL, FALSE, NULL);

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

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

Sem\_m[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;

}