Лабораторная работа №2

по курсу:

«Паралельные и распределённые вычисления»

Тема: «Win32»

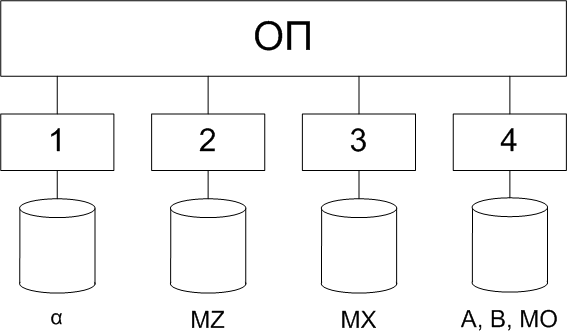
Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

Гульченко \_\_\_\_\_\_

**Техническое задание**

A = (B \* MO)(MX \* MZ \* α)



**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. CH = B \* MOH

**Общие ресурсы:** B

1. MCH = MX \* MZH \* α

**Общие ресурсы:** MX, α

1. AH = C \* MCH

**Общие ресурсы:** C

**Этап 2. Разработка алгоритмов процессов (задач)**

**№ Т1 ТС/КУ**

1. Ввод α
2. Сигнал задачам Т2, Т3, Т4 о конце ввода S2.1, S3.1, S4.1
3. Ждать введения MO, B в Т4 W4.1
4. Копирование B1 := B КУ
5. Cчёт CH = B1 \* MOH
6. Сигнал задачам Т2, Т3, Т4 о завершении счёта С S2.2, S3.2, S4.2
7. Ждать введения МX в Т3 и МZ в Т2 W2.2, W3.2
8. Копирование МХ1 := MХ, α1 := α КУ
9. Счёт MCH = MХ1 \* MZH \* α1
10. Ждать завершения счёта С в задачах Т2, Т3, Т4 W2.3, W3.3, W4.3
11. Копирование С1 := C КУ
12. Счёт AH = C1 \* MCH
13. Сигнал задаче Т4 о завершении счёта. S4.3

**№ Т2 ТС/КУ**

1. Ввод MZ
2. Сигнал задачам Т1, Т3, Т4 о конце ввода S1.1, S3.1, S4.1
3. Ждать введения MO, B в Т4 W4.1
4. Копирование B2 := B КУ
5. Cчёт CH = B2 \* MOH
6. Сигнал задачам Т1, Т3, Т4 о завершении счёта С S1.2, S3.2, S4.2
7. Ждать введения МX в T3 и α в Т1 W1.2, W3.2
8. Копирование МХ2 := MХ, α2 := α КУ
9. Счёт MCH = MХ2 \* MZH \* α2
10. Ждать завершения счёта С в задачах Т1, Т3, Т4 W1.3, W3.3, W4.3
11. Копирование С2 := C КУ
12. Счёт AH = C2 \* MCH
13. Сигнал задаче Т4 о завершении счёта. S4.3

**№ Т3 ТС/КУ**

1. Ввод MX
2. Сигнал задачам Т1, Т2, Т4 о конце ввода S1.1, S2.1, S4.1
3. Ждать введения MO, B в Т4 W4.1
4. Копирование B3 := B КУ
5. Cчёт CH = B3 \* MOH
6. Сигнал задачам Т1, Т2, Т4 о завершении счёта С S1.2, S2.2, S4.2
7. Ждать введения МZ в T2 и α в Т1 W1.2, W2.2
8. Копирование МХ3 := MХ, α := α КУ
9. Счёт MCH = MХ3 \* MZH \* α3
10. Ждать завершения счёта С в задачах Т1, Т2, Т4 W1.3, W2.3, W4.3
11. Копирование С3 := C КУ
12. Счёт AH = C3 \* MCH
13. Сигнал задаче Т4 о завершении счёта. S4.3

**№ Т4 ТС/КУ**

1. Ввод B, MO
2. Сигнал задачам Т1, Т2, Т3 о конце ввода S1.1, S2.1, S3.1
3. Копирование B4 := B КУ
4. Cчёт CH = B4 \* MOH
5. Сигнал задачам Т1, Т2, Т3 о завершении счёта С S1.2, S2.2, S3.2
6. Ждать введения МZ в T2, α в Т1 и MX в Т3 W1.1, W2.1, W3.1
7. Копирование МХ4 := MХ, α4 := α КУ
8. Счёт MCH = MХ4 \* MZH \* α4
9. Ждать завершения счёта С в задачах Т1, Т2, Т3 W1.2, W2.2, W3.2
10. Копирование С4 := C КУ
11. Счёт AH = C4 \* MCH
12. Ждать завершения счёта А в задачах Т1, Т2, Т3 . W1.3, W2.3, W3.3
13. Вывод А.

**Этап 3. Разработка схемы взаимодействия задач**



**Этап 4. Разработка программы**

/\*\*

\* @author IO-83

\* Paralel and distributed computing

\* Laboratory work #2. Win32

\*/

#include "stdafx.h"

#include "windows.h"

#include <iostream>

#include <fstream>

#include <algorithm>

#include <string>

using namespace std;

int N = 4;

const int P = 4;

const int H = N/P;

//========================================================

int alfa;

int \* A = new int[N];

int \* B = new int[N];

int \* C = new int[N];

int \*\* MO = new int\*[N];

int \*\* MX = new int\*[N];

int \*\* MZ = new int\*[N];

int \*\* MC = new int\*[N];

//Semaphores

HANDLE S1, S2, S3, S4, S5, S6, S7, S8;

//S1 - (T1) Синхро по вводу

//S2 - (T2) Синхро по вводу

//S3 - (T3) Синхро по вводу

//S4 - (T4) Синхро по вводу

//S5 - (T1) Синхро по счёту С

//S6 - (T2) Синхро по счёту С

//S7 - (T3) Синхро по счёту С

//S8 - (T4) Синхро по счёту С

//Critical section

CRITICAL\_SECTION CS\_B, CS\_MX;

//Mutex

HANDLE Mut;

//Events

HANDLE Eve1, Eve2, Eve3;

static void inputVector(int\* vec) {

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

vec[j] = 1;

}

}

static void inputMatrix(int\*\* matr) {

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

{

matr[i] = new int[N];

}

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

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

matr[i][j] = 1;

}

}

}

//========================================================

void T1()

{

int alfa1, sum;

int\*\* MX1 = new int\*[N];

int\* B1 = new int[N];

int\* C1 = new int[N];

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

{

MX1[i] = new int[N];

}

int task = 1; //Task number

cout << "T1: started" << endl;

//1. Ввод α

alfa = 1;

//2. Сигнал задачам Т2, Т3, Т4 о конце ввода

ReleaseSemaphore(S1, 3, NULL); //S2.1, S3.1, S4.1

//3. Ждать введения MO, B в Т4

WaitForSingleObject(S4, INFINITE); //W4.1

cout << "LOL";

//4. Копирование B1 := B (КРИТИЧЕСКАЯ СЕКЦИЯ)

EnterCriticalSection(&CS\_B);

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

B1[i] = B[i];

LeaveCriticalSection(&CS\_B);

//5. Cчёт CH = B1 \* MOH

for (int i = (task - 1) \* H; i < task \* H; i++) {

sum = 0;

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

sum = sum + B1[j] \* MX[j][i];

}

C[i] = sum;

}

//6. Сигнал задачам Т2, Т3, Т4 о завершении счёта С

ReleaseSemaphore(S5, 3, NULL); //S2.2 S3.2 S4.2

//7. Ждать введения МX в Т3 и МZ в Т2

WaitForSingleObject(S3, INFINITE); //W3.2

WaitForSingleObject(S2, INFINITE); //W2.2

//8. Копирование МХ1 := MХ, α1 := α

EnterCriticalSection(&CS\_MX);

alfa1 = alfa;

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

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

MX1[i][j] = MX[i][j];

LeaveCriticalSection(&CS\_MX);

//9. Счёт MCH = MХ1 \* MZH \* α1

for (int z = (task - 1) \* H; z < task \* H; z++){

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

sum = 0;

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

sum = sum + MZ[z][j] \* MX1[j][i];

}

MC[z][i] = sum \* alfa1;

}

}

//10. Ждать завершения счёта С в задачах Т2, Т3, Т4

WaitForSingleObject(S6, INFINITE); //W2.3

WaitForSingleObject(S7, INFINITE); //W3.3

WaitForSingleObject(S8, INFINITE); //W4.3

//11. Копирование С1 := C

WaitForSingleObject(Mut, INFINITE);

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

C1[i] = C[i];

ReleaseMutex(Mut);

//12. Счёт AH = C1 \* MCH

for (int i = (task - 1) \* H; i < task \* H; i++) {

sum = 0;

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

sum = sum + C1[j] \* MC[i][j];

}

A[i] = sum;

}

//13. Сигнал задаче Т4 о завершении счёта.

SetEvent(Eve1); //S4.3

cout << "T1 finished" << endl;

}

//========================================================

void T2()

{

int alfa2, sum;

int\*\* MX2 = new int\*[N];

int\* B2 = new int[N];

int\* C2 = new int[N];

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

{

MX2[i] = new int[N];

}

int task = 2; //Task number

cout << "T2: started" << endl;

//1. Ввод МZ

inputMatrix(MZ);

//2. Сигнал задачам Т1, Т3, Т4 о конце ввода

ReleaseSemaphore(S2, 3, NULL); //S1.1, S3.1, S4.1

//3. Ждать введения MO, B в Т4

WaitForSingleObject(S4, INFINITE); //W4.1

//4. Копирование B2 := B

EnterCriticalSection(&CS\_B);

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

B2[i] = B[i];

LeaveCriticalSection(&CS\_B);

//5. Cчёт CH = B2 \* MOH

for (int i = (task - 1) \* H; i < task \* H; i++) {

sum = 0;

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

sum = sum + B2[j] \* MX[j][i];

}

C[i] = sum;

}

//6. Сигнал задачам Т1, Т3, Т4 о завершении счёта С

ReleaseSemaphore(S6, 3, NULL); //S1.2, S3.2, S4.2

//7. Ждать введения МX в T3 и α в Т1 W1.2, W3.2

WaitForSingleObject(S1, INFINITE); //W1.2

WaitForSingleObject(S3, INFINITE); //W3.2

//8. Копирование МХ2 := MХ, α2 := α

EnterCriticalSection(&CS\_MX);

alfa2 = alfa;

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

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

MX2[i][j] = MX[i][j];

LeaveCriticalSection(&CS\_MX);

//9. Счёт MCH = MХ2 \* MZH \* α2

for (int z = (task - 1) \* H; z < task \* H; z++){

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

sum = 0;

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

sum = sum + MZ[z][j] \* MX2[j][i];

}

MC[z][i] = sum \* alfa2;

}

}

//10. Ждать завершения счёта С в задачах Т1, Т3, Т4

WaitForSingleObject(S5, INFINITE); //W1.3

WaitForSingleObject(S7, INFINITE); //W3.3

WaitForSingleObject(S8, INFINITE); //W4.3

//11. Копирование С2 := C

WaitForSingleObject(Mut, INFINITE);

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

C2[i] = C[i];

ReleaseMutex(Mut);

//12. Счёт AH = C2 \* MCH

for (int i = (task - 1) \* H; i < task \* H; i++) {

sum = 0;

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

sum = sum + C2[j] \* MC[i][j];

}

A[i] = sum;

}

//13. Сигнал задаче Т4 о завершении счёта.

SetEvent(Eve2); //S4.3

cout << "T2 finished" << endl;

}

//========================================================

void T3()

{

int alfa3, sum;

int\*\* MX3 = NULL;

MX3 = new int\*[N];

int\* B3 = new int[N];

int\* C3 = new int[N];

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

{

MX3[i] = new int[N];

}

int task = 3; //Task number

cout << "T3: started" << endl;

//1. Ввод МX

inputMatrix(MX);

//2. Сигнал задачам Т1, Т2, Т4 о конце ввода

ReleaseSemaphore(S3, 3, NULL); //S1.1, S2.1, S4.1

//3. Ждать введения MO, B в Т4

WaitForSingleObject(S4, INFINITE); //W4.1

//4. Копирование B3 := B

EnterCriticalSection(&CS\_B);

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

B3[i] = B[i];

LeaveCriticalSection(&CS\_B);

//5. Cчёт CH = B3 \* MOH

for (int i = (task - 1) \* H; i < task \* H; i++) {

sum = 0;

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

sum = sum + B3[j] \* MX[j][i];

}

C[i] = sum;

}

//6. Сигнал задачам Т1, Т2, Т4 о завершении счёта С

ReleaseSemaphore(S7, 3, NULL); //S1.2, S2.2, S4.2

//7. Ждать введения МZ в T2 и α в Т1

WaitForSingleObject(S1, INFINITE); //W1.2

WaitForSingleObject(S2, INFINITE); //W2.2

//8. Копирование МХ3 := MХ, α := α

EnterCriticalSection(&CS\_MX);

alfa3 = alfa;

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

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

MX3[i][j] = MX[i][j];

LeaveCriticalSection(&CS\_MX);

//9. Счёт MCH = MХ3 \* MZH \* α3

for (int z = (task - 1) \* H; z < task \* H; z++){

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

sum = 0;

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

sum = sum + MZ[z][j] \* MX3[j][i];

}

MC[z][i] = sum \* alfa3;

}

}

//10. Ждать завершения счёта С в задачах Т1, Т2, Т4

WaitForSingleObject(S5, INFINITE); //W1.3

WaitForSingleObject(S6, INFINITE); //W2.3

WaitForSingleObject(S8, INFINITE); //W4.3

//11. Копирование С3 := C

WaitForSingleObject(Mut, INFINITE);

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

C3[i] = C[i];

ReleaseMutex(Mut);

//12. Счёт AH = C3 \* MCH

for (int i = (task - 1) \* H; i < task \* H; i++) {

sum = 0;

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

sum = sum + C3[j] \* MC[i][j];

}

A[i] = sum;

}

//13. Сигнал задаче Т4 о завершении счёта.

SetEvent(Eve3); //S4.3

cout << "T3 finished" << endl;

}

//========================================================

void T4()

{

int alfa4, sum;

int\*\* MX4 = new int\*[N];

int\* B4 = new int[N];

int\* C4 = new int[N];

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

{

MX4[i] = new int[N];

}

int task = 4; //Task number

cout << "T4: started" << endl;

//1. Ввод B, MO

inputMatrix(MO);

inputVector(B);

//2. Сигнал задачам Т1, Т2, Т3 о конце ввода

ReleaseSemaphore(S4, 3, NULL); //S1.1, S2.1, S3.1

//3. Копирование B4 := B

EnterCriticalSection(&CS\_B);

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

B4[i] = B[i];

LeaveCriticalSection(&CS\_B);

//4. Cчёт CH = B4 \* MOH

for (int i = (task - 1) \* H; i < task \* H; i++) {

sum = 0;

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

sum = sum + B4[j] \* MO[j][i];

}

C[i] = sum;

}

//5. Сигнал задачам Т1, Т2, Т3 о завершении счёта С

ReleaseSemaphore(S8, 3, NULL); //S1.2, S2.2, S3.2

//6. Ждать введения МZ в T2, α в Т1 и MX в Т3

WaitForSingleObject(S1, INFINITE); //W1.2

WaitForSingleObject(S2, INFINITE); //W2.2

WaitForSingleObject(S3, INFINITE); //W3.2

//7. Копирование МХ4 := MХ, α4 := α

EnterCriticalSection(&CS\_MX);

alfa4 = alfa;

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

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

MX4[i][j] = MX[i][j];

LeaveCriticalSection(&CS\_MX);

//8. Счёт MCH = MХ4 \* MZH \* α4

for (int z = (task - 1) \* H; z < task \* H; z++){

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

sum = 0;

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

sum = sum + MZ[z][j] \* MX4[j][i];

}

MC[z][i] = sum \* alfa4;

}

}

//9. Ждать завершения счёта С в задачах Т1, Т2, Т3

WaitForSingleObject(S5, INFINITE); //W1.3

WaitForSingleObject(S6, INFINITE); //W2.3

WaitForSingleObject(S7, INFINITE); //W3.3

//10. Копирование С4 := C

WaitForSingleObject(Mut, INFINITE);

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

C4[i] = C[i];

ReleaseMutex(Mut);

//11. Счёт AH = C4 \* MCH

for (int i = (task - 1) \* H; i < task \* H; i++) {

sum = 0;

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

sum = sum + C4[j] \* MC[i][j];

}

A[i] = sum;

}

//12. Ждать завершения счёта А в задачах Т1, Т2, Т3

WaitForSingleObject(Eve1, NULL);//W1.3

WaitForSingleObject(Eve2, NULL);//W2.3

WaitForSingleObject(Eve3, NULL);//W3.3

//13. Вывод А.

cout << " " << endl;

cout << " " << endl;

cout << " " << endl;

cout << " " << endl;

if( N < 12 ) {

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

cout << A[i] << " ";

}

}

cout << "T4 finished" << endl;

}

//========================================================

int main()

{

DWORD Tid1, Tid2, Tid3, Tid4;

HANDLE Thread1, Thread2, Thread3, Thread4;

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

{

MC[i] = new int[N];

}

Eve1 = CreateEvent(NULL, 0, 1, NULL);

Eve2 = CreateEvent(NULL, 0, 1, NULL);

Eve3 = CreateEvent(NULL, 0, 1, NULL);

S1 = CreateSemaphore(NULL, 0, 3, NULL);

S2 = CreateSemaphore(NULL, 0, 3, NULL);

S3 = CreateSemaphore(NULL, 0, 3, NULL);

S4 = CreateSemaphore(NULL, 0, 3, NULL);

S5 = CreateSemaphore(NULL, 0, 3, NULL);

S6 = CreateSemaphore(NULL, 0, 3, NULL);

S7 = CreateSemaphore(NULL, 0, 3, NULL);

S8 = CreateSemaphore(NULL, 0, 3, NULL);

Mut = CreateMutex(NULL, FALSE, NULL);

InitializeCriticalSection(&CS\_B);

InitializeCriticalSection(&CS\_MX);

size\_t st = 50 \* 1024 \* 1024;

Thread1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE) T1, NULL, 0, &Tid1);

Thread2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE) T2, NULL, 0, &Tid2);

Thread3 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE) T3, NULL, 0, &Tid3);

Thread4 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE) T4, NULL, 0, &Tid4);

CloseHandle(Thread1);

CloseHandle(Thread2);

CloseHandle(Thread3);

cin.get();

return 0;

}