МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Навчально-науковий Інститут

Комп’ютерних Інформаційних технологій

Кафедра комп’ютеризованих систем управління

Лабораторна робота№4

З дисципліни “Паралельні та розподілені обчислення”

На тему “ **Програмування для комп’ютерних систем зі загальною пам’яттю** ”

Виконав:

Студент ННІКІТ СП-424

Єлістраток К.Є

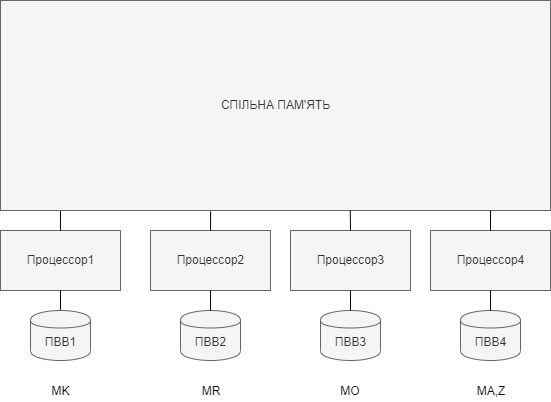
Перевірив:

Корочкін О В

Київ 2019

**ЗАВДАННЯ НА РОБОТУ**

**1. Структура паралельної комп’ютерної системи**



1. **Математична задача**

MA = min(Z)\*MO + max(Z) \* (MR \* MK)

1. **Мова програмування:** JAVA
2. **Засоби програмування взаємодії потоків (задач)** : МОНІТОРИ

**ВИКОНАННЯ РОБОТИ**

**Етап 1. Побудова паралельного алгоритму**

1. xi = min(Zh), i = 1,4
2. x = min(x,xi), i = 1,4
3. ai = max(Zh), i = 1,4
4. a = max(x,xi), i = 1,4
5. MUh = (MRh \* MK)
6. Ah = x \* MOh + a \* MUh

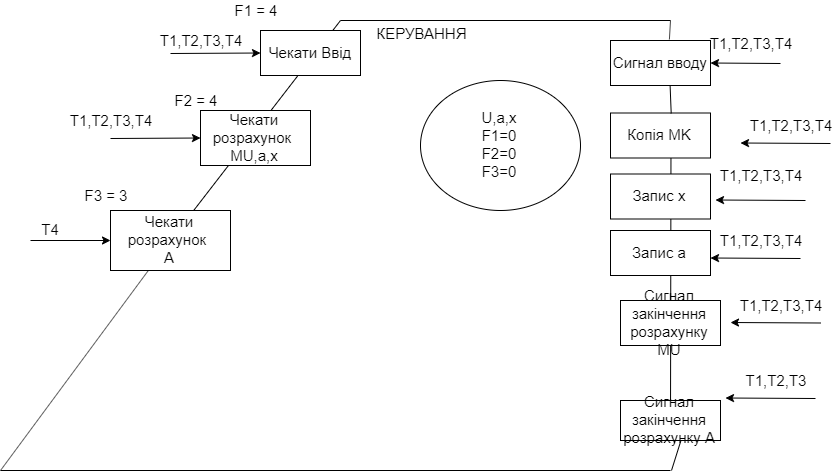
**Етап 2. Розроблення алгоритмів потоків (задач) .**

|  |  |  |
| --- | --- | --- |
| **№** | **Задача *Т1*** | ***Точки синхронізації*** |
| **1** | Введення MK |  |
| **2** | Сигнал T2,T3,T4 про ввід MK | S12,S13,S14 |
| **3** | Очікування сигналу T2, T3,T4 про ввід | W12,W13,W1,4 |
| **4** | MK1 = MK | КД |
| **5** | x1 = min(Zh) |  |
| **6** | a1 = max(Zh) |  |
| **7** | a = max(a,a1) | КД |
| **8** | x = min(x,x1) | КД |
| **5** | Uh = (MRh \* MK1) |  |
| **6** | Сигнал T2,T3,T4 про завершення розрахунку U | S15,S16,S17 |
| **7** | Очікування сигналу T2,T3,T4 про завершення розрахунку U | W15,W16,W17 |
| **8** | A = x \* MOh + a \* MUh |  |
| **9** | Сигнал T4 про завершення розрахунку Ah | S1,8 |

|  |  |  |
| --- | --- | --- |
| **№** | **Задача *Т2*** | ***Точки синхронізації*** |
| **1** | Введення MR |  |
| **2** | Сигнал T2,T3,T4 про ввід MR | S2,1 , S2,3 , S2,4 |
| **3** | Очікування сигналу T2, T3,T4 про ввід | W2,1 , W2,3 , W2,4 |
| **4** | MK2 = MK | КД |
| **5** | x2 = min(Zh) |  |
| **6** | a2 = max(Zh) |  |
| **7** | a = max(a,a2) | КД |
| **8** | x = min(x,x2) | КД |
| **5** | Uh = (MRh \* MK2) |  |
| **6** | Сигнал T2,T3,T4 про завершення розрахунку U | S25,S26,S27 |
| **7** | Очікування сигналу T2,T3,T4 про завершення розрахунку U | W25,W26,W27 |
| **8** | A = x \* MOh + a \* MUh |  |
| **9** | Сигнал T4 про завершення розрахунку Ah | S2,9 |

|  |  |  |
| --- | --- | --- |
| **№** | **Задача *Т3*** | ***Точки синхронізації*** |
| **1** | Введення MK |  |
| **2** | Сигнал T2,T3,T4 про ввід MK | S3,1 , S3,2, S3,4 |
| **3** | Очікування сигналу T2, T3,T4 про ввід | W3,1 , W3,2 , W3,4 |
| **4** | MK3 = MK | КД |
| **5** | x3 = min(Zh) |  |
| **6** | a3 = max(Zh) |  |
| **7** | a = max(a,a3) | КД |
| **8** | x = min(x,x3) | КД |
| **5** | Uh = (MRh \* MK3) |  |
| **6** | Сигнал T2,T3,T4 про завершення розрахунку U | S3,5 , S3,6 , S3,7 |
| **7** | Очікування сигналу T2,T3,T4 про завершення розрахунку U | W35,W36,W37 |
| **8** | A = x \* MOh + a \* MUh |  |
| **9** | Сигнал T4 про завершення розрахунку Ah | S3,10 |
| **№** | **Задача *Т4*** | ***Точки синхронізації*** |
| **1** | Введення MK |  |
| **2** | Сигнал T2,T3,T4 про ввід MK | S4,1 , S4,2 , S4,3 |
| **3** | Очікування сигналу T2, T3,T4 про ввід | W4,1 , W4,2 , W4,3 |
| **4** | MK4 = MK | КД |
| **5** | x4 = min(Zh) |  |
| **6** | a4 = max(Zh) |  |
| **7** | a = max(a,a4) | КД |
| **8** | x = min(x,x4) | КД |
| **5** | Uh = (MRh \* MK4) |  |
| **6** | Сигнал T2,T3,T4 про завершення розрахунку U | S4,5 , S4,6 , S4,7 |
| **7** | Очікування сигналу T2,T3,T4 про завершення розрахунку U | W4,5 , W4,6 , W4,7 |
| **8** | A = x \* MOh + a \* MUh |  |
| **9** | Очікування сигналу про розрахунки A від T1,T2,T3 | W4,8 W4,9 W4,10 |
| **10** | Вивід A |  |

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



*//Лаб. Робота №4 Java. Мониторы*

*// Єлістратова К.Є*

*// Групи СП-424*

*// 01.12.2019*

*Monitor.java*

*package kirill\_monytor;*

*import java.util.concurrent.atomic.AtomicInteger;*

*public class Monitor {*

*private Object S\_monitor = new Object();*

*private Object a\_monitor = new Object();*

*private AtomicInteger calc\_monitor = new AtomicInteger(Lab4.P - 1);*

*private int a = 0;*

*private int x = 100;*

*private AtomicInteger countInput;*

*private AtomicInteger countU;*

*private AtomicInteger countA;*

*// INITIALIZE*

*public Monitor(AtomicInteger countInput, AtomicInteger countU, AtomicInteger countA) {*

*this.countInput = countInput;*

*this.countU = countU;*

*this.countMultiply = countMultiply;*

*}*

*// END INITIALIZE*

*public synchronized int[][] dublicateMK(int MK) {*

*synchronized (a\_monitor) {*

*return MK;*

*}*

*}*

*//MAX Z*

*public synchronized void addToA(int i) {*

*(a\_monitor) {*

*if(i > a){*

*a = i;*

*}*

*}*

*}*

*//END MAX Z*

*//MIN Z*

*public synchronized void addToX(int i) {*

*(a\_monitor) {*

*if(i < x){*

*x = i;*

*}*

*}*

*}*

*//END MIN Z*

*//SYNCHR TO INPUT*

*public synchronized void wait\_Input() {*

*(countInput) {*

*try {*

*while (countInput.get() > 0) {*

*countInput.wait();*

*}*

*} catch (InterruptedException e) {*

*e.printStackTrace();*

*}*

*}*

*}*

*public synchronized void signal\_Input() {*

*(countInput) {*

*if (countInput.decrementAndGet() == 0) {*

*countInput.notifyAll();*

*}*

*}*

*}*

*//END SYNCHR TO INPUT*

*// SYNCHR FOR CALC U*

*public synchronized void wait\_U() {*

*(countU) {*

*try {*

*while (countU.get() > 0) {*

*countU.wait();*

*}*

*} catch (InterruptedException e) {*

*e.printStackTrace();*

*}*

*}*

*}*

*public synchronized void signal\_U() {*

*(countU) {*

*if (countU.decrementAndGet() == 0) {*

*countU.notifyAll();*

*}*

*}*

*}*

*//END SYNCHR FOR CALC U*

*// SYNCHR FOR CALC U*

*public synchronized void wait\_A() {*

*(countA) {*

*try {*

*while (countA.get() > 0) {*

*countA.wait();*

*}*

*} catch (InterruptedException e) {*

*e.printStackTrace();*

*}*

*}*

*}*

*public synchronized void signal\_U() {*

*(countA) {*

*if (countA.decrementAndGet() == 0) {*

*countA.notifyAll();*

*}*

*}*

*}*

*//END SYNCHR FOR CALC U*

*}*

*LAB4.JAVA*

*// Лаб. Робота №4*

*//Java. Використання моніторів*

*//Єлістратов К.Є*

*// 01.12.2019*

*//Lab4.java*

*package kirill\_monytor;*

*import java.util.Arrays;*

*import java.util.concurrent.atomic.AtomicInteger;*

*public class Lab4 {*

*//INIT OF DATA*

*static final int N = 8;*

*static final int P = 4;*

*static final int H = N / P;*

*static int Z[] = new int[N];*

*static int MK[][] = new int[N][N];*

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

*static int MR[][] = new int[N][N];*

*static int MA[][] = new int[N][N];*

*static int MU[][] = new int[N][N];*

*// END INIT OF DATA*

*public static void main(String[] args) {*

*Monitor monitor = new Monitor(new AtomicInteger(P), new AtomicInteger(P), new AtomicInteger(P)); //4 thread to input 4 to calculate 4 to output*

*//T1*

*new Thread(() -> {*

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

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

*MK[i][j] = 1;*

*}*

*}*

*monitor.signal\_Input();*

*monitor.wait\_Input();*

*for (int i = 0; i < h; i++) {*

*monitor.addA(Z[i])*

*monitor.addX(Z[i])*

*}*

*int [][] MK1 = monutor.dublicateMK();*

*for (int i = 0; i < h; i++) {*

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

*MU[i][j] = MR[i][j] \* MK1[j][i];*

*}*

*}*

*monitor.signal\_U();*

*monitor.wait\_U();*

*for(int i = 0;i<h;i++){*

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

*MA[i][j] = x \* MO[i][j] + a \* MU[i][j];*

*}*

*}*

*monitor.signal\_A();*

*})*

*.start();*

*//T2*

*new Thread(() -> {*

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

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

*MR[i][j] = 1;*

*}*

*}*

*monitor.signal\_Input();*

*monitor.wait\_Input();*

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

*monitor.addA(Z[i])*

*monitor.addX(Z[i])*

*}*

*int [][] MK2 = monutor.dublicateMK();*

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

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

*MU[i][j] = MR[i][j] \* MK2[j][i];*

*}*

*}*

*monitor.signal\_U();*

*monitor.wait\_U();*

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

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

*MA[i][j] = x \* MO[i][j] + a \* MU[i][j];*

*}*

*}*

*monitor.signal\_A();*

*}) // END T2*

*.start();*

*//T3*

*new Thread(() -> {*

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

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

*MO[i][j] = 1;*

*}*

*}*

*monitor.signal\_Input();*

*monitor.wait\_Input();*

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

*monitor.addA(Z[i])*

*monitor.addX(Z[i])*

*}*

*int [][] MK3 = monutor.dublicateMK();*

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

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

*MU[i][j] = MR[i][j] \* MK3[j][i];*

*}*

*}*

*monitor.signal\_U();*

*monitor.wait\_U();*

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

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

*MA[i][j] = x \* MO[i][j] + a \* MU[i][j];*

*}*

*}*

*monitor.signal\_A();*

*}) // END T3*

*.start();*

*//T4*

*new Thread(() -> {*

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

*Z[i] = 1;*

*}*

*monitor.signal\_Input();*

*monitor.wait\_Input();*

*for (int i = 0; i < h; i++) {*

*monitor.addA(Z[i])*

*monitor.addX(Z[i])*

*}*

*int [][] MK4 = monutor.dublicateMK();*

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

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

*MU[i][j] = MR[i][j] \* MK4[j][i];*

*}*

*}*

*monitor.signal\_U();*

*monitor.wait\_U();*

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

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

*MA[i][j] = x \* MO[i][j] + a \* MU[i][j];*

*}*

*}*

*monitor.wait\_A();*

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

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

*System.out.println(MA[i][j]);*

*}*

*}*

*}) // END T4*

*.start();*

*}*

*}*