Национальный технический университет

«Киевский политехнический институт»

Факультет информатики и вычислительной техники

## Кафедра Вычислительной Техники

#### Лабораторная работа №3

По курсу: Параллельное программирование

студента III-го курса,

группы ИВ-92

Петрука В.О.

## Киев 2011 г.

//Laboratory work 3

//Petruk Vadim, IO-92

//Variant: 1.13 C = A - B + D

// 2.28 MD = MIN(MA)\*MB\*MC

// 3.14 D = (A + B)\*(MA - MB)

**public** **class** Lab3 {

**private** **static** **int** *N* = 4;

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

Runnable func1 = **new** Func1(*N*);

Thread thread1 = **new** Thread(func1,"Func1");

thread1.setPriority(4);

Runnable func2 = **new** Func2(*N*);

Thread thread2 = **new** Thread(func2,"Func2");

thread2.setPriority(7);

Runnable func3 = **new** Func3(*N*);

Thread thread3 = **new** Thread(func3,"Func3");

thread3.setPriority(1);

thread1.start();

thread2.start();

thread3.start();

}

}

**public** **class** InputOutputClass {

**public** **int**[] VectorIn(**int** N){

**int**[] X = **new** **int**[N];

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

X[i] = 1;

}

**return** X;

}

**public** **void** VectorOut(**int**[] X){

**for**(**int** i=0; i<X.length; i++){

System.*out*.print(X[i]+" ");

}

System.*out*.println();

}

**public** **int**[][] MatrixIn(**int** N){

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

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

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

MX[i][j] = 1;

}

}

**return** MX;

}

**public** **void** MatrixOut(**int**[][] MX){

**for** (**int** i=0; i<MX.length; i++){

**for**(**int** j=0; j < MX[0].length; j++){

System.*out*.print(MX[i][j]+" ");

**if** (j==MX[0].length-1)

System.*out*.println();

}

}

}

}

// Func1: C := A - B + D

**public** **class** Func1 **implements** Runnable{

**private** **int** N;

**private** **int**[] A;

**private** **int**[] B;

**private** **int**[] C;

**private** **int**[] D;

**private** InputOutputClass \_InOutVar= **new** InputOutputClass();

**public** Func1(**int** N){

**this**.N = N;

A = **new** **int**[N];

B = **new** **int**[N];

C = **new** **int**[N];

D = **new** **int**[N];

}

**public** **void** run(){

System.*out*.println("Func1 is started");

A = \_InOutVar.VectorIn(N);

B = \_InOutVar.VectorIn(N);

D = \_InOutVar.VectorIn(N);

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

C[i] = A[i] - B[i] + D[i];

}

**if** (N<=8){

\_InOutVar.VectorOut(C);

}

System.*out*.println("Func1 is finished");

}

}

// Func2: MD := MIN(MA)\*MB\*MC

**public** **class** Func2 **implements** Runnable{

**private** **int** N;

**private** **int**[][] MA;

**private** **int**[][] MB;

**private** **int**[][] MC;

**private** **int**[][] MD;

**private** InputOutputClass \_InOutVar= **new** InputOutputClass();

**public** Func2(**int** N){

**this**.N = N;

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

MB = **new** **int**[N][N];

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

MD = **new** **int**[N][N];

}

**public** **void** run(){

System.*out*.println("Func2 is started");

MA = \_InOutVar.MatrixIn(N);

MB = \_InOutVar.MatrixIn(N);

MC = \_InOutVar.MatrixIn(N);

//MD = 0

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

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

MD[i][j]=0;

}

}

//MD := MB\*MC

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

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

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

MD[i][j] = MD[i][j]+MB[i][k]\*MC[k][j];

}

}

}

//search MIN(MA)

**int** min\_MA = MA[0][0];

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

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

**if** (MA[i][j] < min\_MA){

min\_MA = MA[i][j];

}

}

}

//MD := MIN(MA)\*MB\*MC

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

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

MD[i][j]=MD[i][j]\*min\_MA;

}

}

**if** (N<=8){

\_InOutVar.MatrixOut(MD);

}

System.*out*.println("Func2 is finished");

}

}

//Func3: D := (A + B)\*(MA - MB)

**public** **class** Func3 **implements** Runnable{

**private** **int** N;

**private** **int**[] A;

**private** **int**[] B;

**private** **int**[][] MA;

**private** **int**[][] MB;

**private** **int**[] D;

**private** InputOutputClass \_InOutVar= **new** InputOutputClass();

**public** Func3(**int** N){

**this**.N = N;

A = **new** **int**[N];

B = **new** **int**[N];

D = **new** **int**[N];

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

MB = **new** **int**[N][N];

}

**public** **void** run(){

System.*out*.println("Func3 is started");

A = \_InOutVar.VectorIn(N);

B = \_InOutVar.VectorIn(N);

MA = \_InOutVar.MatrixIn(N);

MB = \_InOutVar.MatrixIn(N);

**int** C[] = **new** **int**[N];

//C:=A + B

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

C[i] = A[i]+B[i];

}

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

//MC:=MA - MB

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

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

MC[i][j] = MA[i][j] - MB[i][j];

}

}

//MD = 0

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

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

D[i]=0;

}

}

//D := C \* MC

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

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

D[j] = D[j] + C[j]\*MC[j][i];

}

}

**if** (N<=8){

\_InOutVar.VectorOut(D);

}

System.*out*.println("Func3 is finished");

}

}