

## Лабораторная работа №7

По курсу: **Параллельное программирование**

студента III-го курса,  
группы ИВ-93  
Свинарчука С.В.

Киев 2011 г.

```

/*
Laboratorna robota №7
Svinarchuk Sergey, IO-93
variant:
1.24:  $E = A + C - B + D$ 
2.03:  $MC = MA * MB$ 
3.02:  $B = (MC * MB) * A$ 
*/

#include <windows.h>
#include <stdio.h>
#include <mpi.h>

const int size = 5000;
const int value = 1;
void VectorIn(int Vector[size]){
    for(int i=0; i<size; i++){
        Vector[i] = value;
    }
}

void MatrixIn(int Matrix[size][size]){
    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){
            Matrix[i][j] = value;
        }
    }
}

void VectorOut(int Vector[size]){
    for(int i=0; i<size; i++){
        printf("%d ", Vector[i]);
    }
    printf("\n");
}

void MatrixOut(int Matrix[size][size]){
    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){
            printf("%d ", Matrix[i][j]);
        }
        printf("\n");
    }
}

void Func1(void){
    printf("Start thread 1\n");
    int A[size], B[size], C[size], D[size], E[size];
    VectorIn(A);
    VectorIn(B);
    VectorIn(C);
    VectorIn(D);
    for(int i=0; i<size; i++){
        E[i] = A[i]+C[i]-B[i]+D[i];
    }
    if(size<=8){
        VectorOut(E);
    }
    printf("Finish thread 1\n");
}

void Func2(void){
    printf("Start thread 2\n");
    int MA[size][size], MB[size][size], MC[size][size];
    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){
            MC[i][j] = 0;
        }
    }
    MatrixIn(MA);
    MatrixIn(MB);

```

```

    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){
            for(int k=0; k<size; k++){
                MC[i][j] += MA[i][k]*MB[k][j];
            }
        }
    }
    if(size<=8){
        MatrixOut(MC);
    }
    printf("Finish thread 2\n");
}

void Func3(void){
    printf("Start thread 3\n");
    int A[size], B[size], MC[size][size], MB[size][size], MX[size][size];
    VectorIn(A);
    MatrixIn(MC);
    MatrixIn(MB);
    for(int i=0; i<size; i++){
        B[i] = 0;
    }
    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){
            MX[i][j] = 0;
        }
    }
    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){
            for(int k=0; k<size; k++){
                MX[i][j] += MB[i][k]*MC[k][j];
            }
        }
    }
    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){
            B[i] += MX[i][j]*A[j];
        }
    }
    if(size<=8){
        VectorOut(B);
    }
    printf("Finish thread 3\n");
}

int main(int args, char* argvs[]){
    int rank;
    MPI_Init(&args, &argvs);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    if(rank == 0 ){
        Func1();
    }
    if(rank == 1 ){
        Func2();
    }
    if(rank == 2 ){
        Func3();
    }
    MPI_Finalize();
    system("pause");
    return 0;
}

```