#define N 4

#include <stdio.h>

#include <math.h>

#include <sys/time.h>

#include <stdlib.h>

#include <stddef.h>

#include "mpi.h"

void print\_results(char \*prompt, int a[N][N]);

int main(int argc, char \*argv[])

{

int i, j, k, rank, size, tag = 99, blksz, sum = 0;

int a[N][N]={{1,2,3,4},{5,6,7,8},{9,1,2,3},{4,5,6,7,}};

int b[N][N]={{1,2,3,4},{5,6,7,8},{9,1,2,3},{4,5,6,7,}};

int c[N][N];

int aa[N],cc[N];

MPI\_Init(&argc, &argv);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

//scatter rows of first matrix to different processes

MPI\_Scatter(a, N\*N/size, MPI\_INT, aa, N\*N/size, MPI\_INT,0,MPI\_COMM\_WORLD);

//broadcast second matrix to all processes

MPI\_Bcast(b, N\*N, MPI\_INT, 0, MPI\_COMM\_WORLD);

MPI\_Barrier(MPI\_COMM\_WORLD);

//perform vector multiplication by all processes

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

{

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

{

sum = sum + aa[j] \* b[j][i]; //MISTAKE\_WAS\_HERE

}

cc[i] = sum;

sum = 0;

}

MPI\_Gather(cc, N\*N/size, MPI\_INT, c, N\*N/size, MPI\_INT, 0, MPI\_COMM\_WORLD);

MPI\_Barrier(MPI\_COMM\_WORLD);

MPI\_Finalize();

if (rank == 0) //I\_ADDED\_THIS

print\_results("C = ", c);

}

void print\_results(char \*prompt, int a[N][N])

{

int i, j;

printf ("\n\n%s\n", prompt);

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

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

printf(" %d", a[i][j]);

}

printf ("\n");

}

printf ("\n\n");

}