**Лабораторная работа №2**

**Использование условий при определении параллельных фрагментов**

**Цель:** изучить основные особенности использования условий при определении параллельных фрагментов в OpenMP на примере использования в рамках языка С++.

**Использование параметра if директивы parallel**

**Код:**

#include <omp.h> // заголовочный файл «omp.h»

#include <iostream>

#include <time.h>

using namespace std;

const int NMAX = 500;

const int MMAX = 500;

const int LIMIT = 20000;

void main()

{

int i, j;

int sum;

float a[NMAX][MMAX];

float counter;

srand(time(0));

setlocale(0, "");

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

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

a[i][j] = rand() % 10 + 1;

}

}

// инициализация данных

clock\_t start = clock();

#pragma omp parallel shared(a) private(i) if (LIMIT < NMAX\*MMAX)

{

#pragma omp for private(j,sum)

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

{

counter = 0;

for (j = 0; j < NMAX - 1; j++) {

sum = 0;

sum = a[i][j] + a[i][j + 1];

while (sum > 0) {

if ((sum % 10) == 7) {

counter++;

}

sum /= 10;

}

}

cout << "\nNumber of 7 in row " << i << " : " << counter << endl;

}

} /\* Завершение параллельного фрагмента \*/

clock\_t end = clock();

if (LIMIT > NMAX \* MMAX) {

printf("LIMIT is unreached \_\_\_\_\_\_\_\_\_\_Time without MP\_\_\_\_\_\_\_\_\_: %f", difftime(end, start));

}

else

printf("\_\_\_\_\_\_\_\_\_\_Time with MP\_\_\_\_\_\_\_\_\_: %f", difftime(end, start));

start = clock();

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

{

counter = 0;

for (j = 0; j < NMAX - 1; j++) {

sum = 0;

sum = a[i][j] + a[i][j + 1];

while (sum > 0) {

if ((sum % 10) == 7) {

counter++;

}

sum /= 10;

}

}

cout << "\nNumber of 7 in row " << i << " : " << counter << endl;

}

/\* Завершение параллельного фрагмента \*/

end = clock();

printf("\_\_\_\_\_\_\_\_\_\_Time without MP\_\_\_\_\_\_\_\_\_: %f", difftime(end, start));

}

Регион выполняется параллельно только при выполнении условия. Иначе – выполняется последовательно. Используется для того, что бы получить наибольшую скорость работы программы. Как на малом количестве данных, когда МП-программа проигрывает последовательной в скорости работы, так и на большом.