

### 3. Технология программирования MPI+OpenMP

#### **Задание 20. Проект в среде Visual Studio 2010 с поддержкой MPI и OpenMP**

Создайте проект в среде Visual Studio 2010 с минимальным кодом с поддержкой MPI и OpenMP.

#### **Задание 21. Программа «I am»**

Напишите программу, в которой в каждом процессе создается  $n$  нитей. Каждая нить должна выводить на экран свой номер, номер родительского процесса и общее количество нитей во всех процессах в следующем формате: I am <Номер нити> thread from <Номер родительского процесса> process. Number of hybrid threads = <Количество нитей \* Количество процессов>.

**Выходные данные:** строки вида «I am <Номер нити> thread from <Номер родительского процесса> process. Number of hybrid threads = <Количество нитей \* Количество процессов>».

**Входные данные:** целое число  $n$  – количество нитей, которые должны быть запущены.

#### **Пример входных и выходных данных для 2-х процессов**

Входные данные	Выходные данные
3	I am 0 thread from 0 process. Number of hybrid threads = 6
	I am 1 thread from 0 process. Number of hybrid threads = 6
	I am 2 thread from 0 process. Number of hybrid threads = 6
	I am 0 thread from 1 process. Number of hybrid threads = 6
	I am 1 thread from 1 process. Number of hybrid threads = 6
	I am 2 thread from 1 process. Number of hybrid threads = 6

#### **Задание 22. Программа «Число $\pi$ »**

Реализуйте программу вычисления числа  $\pi$  (см. задание 8) с использованием MPI+OpenMP.

**Входные данные:** одно целое число  $N$  (точность вычисления).

**Выходные данные:** одно вещественное число  $\pi$ .

#### **Пример входных и выходных данных для 2-х процессов**

Входные данные	Выходные данные
1000000000	3.14159265

## **Указания к заданию 20. Проект в среде Visual Studio 2010 с поддержкой MPI и OpenMP**

1. Создайте консольное приложение в среде Visual Studio с поддержкой OpenMP (см. указания к заданию 1).
2. Подключите к проекту MPICH (см. указания к заданию 14).
3. Скомпилируйте и запустите приложение.

## **Указания к заданию 21. Программа «I am»**

1. Создайте консольное приложение в среде Visual Studio с поддержкой OpenMP и MPI (см. указания к заданию 20).
2. Получите номер процесса и количество процессов в программе в переменные `rank_mpi` и `size_mpi` соответственно.
3. Между функциями `MPI_Init` и `MPI_Finalize` создайте с помощью OpenMP-директивы `parallel` параллельную область:

```
MPI_Init(...);  
// Операторы  
#pragma omp parallel  
{  
    // Операторы  
}  
// Операторы  
MPI_Finalize();
```

4. Получите номер нити и количество нитей в программе в переменные `rank_omp` и `size_omp` соответственно.
5. Внутри параллельной области сформируйте сообщение для вывода на экран:

```
printf("I am %d thread from %d> process. Number of hybrid  
threads = %d.\n", rank_omp,  
rank_mpi, size_omp*size_mpi);
```

6. Скомпилируйте и запустите ваше приложение.

### **Указания к заданию 22. Программа «Число $\pi$ »**

1. Создайте консольное приложение в среде Visual Studio с поддержкой OpenMP и MPI (см. указания к заданию 20)
2. Напишите программу, вычисляющую число  $\pi$ . Основная вычислительная часть программы – это цикл `for`. Распределите все итерации цикла равными долями по процессам, например, следующим образом:

```
for (i=rank; i<N; i=i+size_mpi)
{
    // Вычисления
}
```

где `size_mpi` – количество MPI-процессов в программе.

3. Внутри процесса распределите итерации по нитям с помощью OpenMP директивы `for`.
4. Скомпилируйте и запустите ваше приложение. Убедитесь, что выдается верный результат.