

Санкт-Петербургский национальный
исследовательский институт информационных
технологий, механики и оптики

Физический факультет



Рабочий протокол и отчет по лабораторной работе №4

**[C++ & UNIX]: ОТЧЁТ ПО ЛАБОРАТОРНОЙ
РАБОТЕ №4**

Группа: Z33431

Студент: Пуговкин Владимир

Преподаватель: Маслов И.Д.

1 Цель работы

Познакомить студента с принципами параллельных вычислений. Составить несколько программ в простейшими вычислительными действиями, чтобы освоить принципы параллельных вычислений (когда одни алгоритмы зависят / не зависят от других).

- Содержание файла process.cpp

```

1  #include <boost/process.hpp>
2  #include <iostream>
3  #include <chrono>
4  #include <boost/asio.hpp>
5  using namespace std;
6  namespace bp = boost::process;
7  namespace ba = boost::asio;
8  int main()
9  {
10     try {
11         ba::io_service ioservice;
12         auto start = chrono::high_resolution_clock::now();
13         bp::child process2("./process2", ioservice, bp::std_out > stdout);
14         bp::child process3("./process3", ioservice, bp::std_out > stdout);
15         cout << "Запущены process2 и process3" << endl;
16         process2.wait();
17         process3.wait();
18         auto end = chrono::high_resolution_clock::now();
19         auto duration = chrono::duration_cast<chrono::microseconds>(end-start).count();
20         cout << "Завершены process2 и process3" << endl;
21         cout<<"Время выполнения программы: "<<duration<<" мкс."<<endl;
22     }
23     catch (const bp::process_error& e)
24     {
25         cerr << "Ошибка при запуске процесса: " << e.what() << endl;
26         return 1;
27     }
28     return 0;
29 }

```

- Содержание файла process2.cpp

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  void func1(int n, double x, double& temp1)
5  {
6      for(int i = 0; i < n; i++)
7      {
8          temp1 += pow(x, 2) - pow(x, 2) + pow(x, 4) - pow(x, 5) + x + x;
9      }
10 }
11 int main ()
12 {
13     double temp=0;
14     func1(1000000, 10, temp);
15     cout<<temp<<endl;
16     return 0;
17 }

```

- Содержание файла process3.cpp

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  ✓ void func2(int n, double x, double& temp2)
5  {
6      for(int i = 0; i < n; i++)
7      {
8          temp2 += x + x;
9      }
10 }
11 ✓ int main ()
12 {
13     double temp2=0;
14     func2(1000000, 10, temp2);
15     cout<<temp2<<endl;
16     return 0;
17 }

```

- Содержание файла solo.cpp

```

1  #include <iostream>
2  #include <cmath>
3  #include <chrono>
4  using namespace std;
5  ✓ void func1(int n, double x, double& temp1)
6  {
7      for(int i = 0; i < n; i++)
8      {
9          temp1 += pow(x, 2) - pow(x, 2) + pow(x, 4) - pow(x, 5) + x + x;
10     }
11 }
12 ✓ void func2(int n, double x, double& temp2)
13 {
14     for(int i = 0; i < n; i++)
15     {
16         temp2 += x + x;
17     }
18 }
19 ✓ int main()
20 {
21     int n;
22     double x;
23     cin >> n >> x;
24     auto start = chrono::high_resolution_clock::now();
25     double temp1 = 0, temp2 = 0;
26     func1(n,x,temp1);
27     func2(n,x,temp2);
28     auto end = chrono::high_resolution_clock::now();
29     cout << temp1 + temp2 << endl;
30     auto duration = chrono::duration_cast<chrono::microseconds>(end - start).count();
31     cout<<"Время выполнения программы: "<<duration<<" мкс."<<endl;
32     return 0;
33 }

```

- Содержание файла threads.cpp

```

1  #include <iostream>
2  #include <cmath>
3  #include <chrono>
4  #include <boost/thread.hpp>
5  #include <boost/process.hpp>
6  #include <boost/ref.hpp>
7  using namespace std;
8  using bt = boost::thread; //не пространство имен, а именно класс boost обращающийся к thread.
9  void func1(int n, double x, double& result)
10 {
11     double temp1=0;
12     for(int i = 0; i < n; i++)
13     {
14         temp1 += pow(x, 2) - pow(x, 2) + pow(x, 4) - pow(x, 5) + x + x;
15     }
16     result += temp1;
17 }
18 void func2(int n, double x, double& result)
19 {
20     double temp2=0;
21     for(int i = 0; i < n; i++)
22     {
23         temp2 += x + x;
24     }
25     result = temp2;
26 }
27 int main()
28 {
29     int n;
30     double x;
31     double temp1 = 0, temp2 = 0;
32     cin >> n >> x;
33     auto start = chrono::high_resolution_clock::now();
34     bt thread1(boost::bind(&func1, n, x, boost::ref(temp1)));
35     bt thread2(boost::bind(&func2, n, x, boost::ref(temp2)));
36     thread1.join();
37     thread2.join();
38     auto end = chrono::high_resolution_clock::now();
39     cout<<"Результат 1: "<<temp1<<endl;
40     cout<<"Результат 2: "<<temp2<<endl;
41     auto duration = chrono::duration_cast<chrono::microseconds>(end - start).count();
42     cout<<"Время выполнения программы: "<<duration<<" мкс."<<endl;
43     return 0;
44 }

```

- process.cpp test time

```

vlapugb@MSI:/usr/local/labitmo/lab_04/src$ ./process
Запущены process2 и process3
2e+07
-8.998e+10
Завершены process2 и process3
Время выполнения программы: 114401 мкс.
vlapugb@MSI:/usr/local/labitmo/lab_04/src$

```

- threads.cpp test time

```

vlapugb@MSI:/usr/local/labitmo/lab_04/src$ ./threads
1000000 10
Результат 1: -8.998e+10
Результат 2: 2e+07
Время выполнения программы: 104226 мкс.
vlapugb@MSI:/usr/local/labitmo/lab_04/src$

```

- solo.cpp test time

```

vlapugb@MSI:/usr/local/labitmo/lab_04/src$ ./solo
1000000 10
-8.996e+10
153625
vlapugb@MSI:/usr/local/labitmo/lab_04/src$

```

2 Вывод

В процессе работы были изучены способы параллельного программирования: через процессоры, через потоки. Из времени работы видно, что ускорение не сильно достигнуто, это связано с тем, что данная постановка задачи слишком простая для использования в параллельном программировании и время на создание потоков и дочерних процессоров сильно влияет на итоговое время, в итоге программа может работать даже медленнее.