

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
ГОУ НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ  
ИМ. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА "Информатика и вычислительная техника"

Дисциплина: "Технология распределенной обработки данных"

Отчёт

по лабораторной работе № 2

Тема: "Разработка распределённой системы обработки данных."  
Вариант №9

Проверил:

Гай В. Е.

Выполнил:

Студент гр. 14-В-2

Балыкин Д. Е.

Нижний Новгород 2016

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

## 1. Требования к работе

Разработанный программный комплекс должен состоять из Сервера и Клиента. Функции сервера: хранение удалённого объекта, предоставляющего доступ к заданиям для обработки и результату обработки. Предусмотреть на сервере возможность одновременного доступа к критической секции кода нескольких клиентов. Критическая секция кода - та, к которой гипотетически одновременно могут обратиться несколько клиентов.

Функции клиента (на сервере хранится список клиентов - эта функция уже предусмотрена исходным кодом библиотеки RemoteBase):

1. Управляющие функции (выполняет только один клиент из всего множества клиентов, выполнение данной функции должно выполняться через вызов методов удалённого объекта (удалённый объект хранится на сервере)):

- Формирование и ведение списка заданий (под ведением понимается удаление уже обработанных и предоставление клиенту задания по запросу);

- Получение, объединение и вывод результатов вычислений (результаты вычислений должны выводиться в каждом клиенте, для этого необходимо проверять окончание обработки всех данных по таймеру; объединение результатов вычисления также можно реализовать с использованием таймера);

- Устанавливает флаг того, что управляющий клиент назначен, на сервере сохраняется идентификатор клиента;

2. Вычислительные функции

- Запрос задания с сервера (клиент должен запросить задание только после того, как эти задания были сформированы);

- Обработка данных;

- Отправка результатов обработки на сервер.

Подп. и дата										
Взам. инв. №										
Инв. № дубл.										
Подп. и дата										
Инв. № подл	Разраб.					Лабораторная работа № 2	Разработка распределённой системы обработки данных	Лит	Лист	Листов
	Пров.								2	11
	Т. контр.							14-B-2		
	Н. контр.									
	Утв.									

2. Выполнение лабораторной работы

2.1. Задание

Разработать алгоритм умножения матрицы  $a$  ( $m * n$ ) элементов на вектор  $b$  ( $n$  элементов) по следующей формуле:

$$\sum_{j=1}^n a_{ij}b_j, 1 \leq i \leq m \tag{2.1}$$

Инв. № подл.					Лист 3
Подп. и дата					Лист 3
Инв. № дубл.					Лист 3
Взам. инв. №					Лист 3
Подп. и дата					Лист 3
Инв. № подл.					Лист 3

Ли	Изм.	№ докум.	Подп.	Дат	Лабораторная работа № 2	Лист 3

### 3. Листинг программы

#### 3.1. Клиент

```
1 using System;
2
3 using System.Runtime.Remoting;
4 using System.Runtime.Remoting.Channels;
5 using System.Runtime.Remoting.Channels.Tcp;
6
7 using Lib;
8
9 namespace Client
10 {
11     class Client
12     {
13         TcpChannel chan;
14         SharedObject obj;
15
16         Task task;
17
18         public Client() {
19             chan = new TcpChannel();
20             ChannelServices.RegisterChannel(chan, false);
21             obj = (SharedObject)Activator.GetObject(typeof(Lib.SharedObject), "tcp://localhost:8080/SharedObject");
22         }
23
24         public int run()
25         {
26             task = obj.GetTask();
27             if (task == null)
28                 return 0;
29             int[,] ATemp;
30             int[] BTemp;
31             int[] C;
32             obj.GetData(task, out BTemp, out ATemp);
33             C = new int[SharedObject.n];
```

Инв. № подл	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
										3
Ли	Изм.	№ докум.	Подп.	Дат	Лабораторная работа № 2					



```

75
76         return 1;
77     }
78     static void Main(string[] args)
79     {
80         Client clientObj = new Client();
81         Console.Out.WriteLine("Клиент запущен");
82
83         while (clientObj.run() != 0)
84             Console.In.ReadLine();
85
86         Console.Out.WriteLine("Задания кончились, нажмите Enter для выхода");
87         Console.ReadLine();
88     }
89 }
90 }

```

### 3.2. Библиотека

```

1 using System;
2 using System.Collections.Generic;
3
4 namespace Lib
5 {
6     public class SharedObject : MarshalByRefObject
7     {
8         static int i;
9         public static int m = 6; //кол-во строк
10        public static int n = 6; //кол-во столбцов
11
12        const int dataCount = 6; //всего строк в матрице
13        const int tasksCount = 6; //максимальное число задач
14
15        Queue<Task> QueTasks; // очередь задач ожидающих обработ
ки
16        Object tasksLock;
17
18        public int[,] A = new int[m, n];    //матрица для умножения
19        public int[] B = new int[n];    //вектор-столбец для умножени
я
20        public int[] C = new int[m];    //хранение результата
21        Object dataLock;

```

[illegible]

```

22
23     public SharedObject() {
24         QueTasks = new Queue<Task>();
25         CreateData();
26         CreateTasks();
27
28         tasksLock = new Object();
29         dataLock = new Object();
30     }
31
32     void CreateTasks() {
33         Log.Print("\n\nСоздание задач...\n");
34         Task temp;
35
36         //распределение массива поровну на каждого клиента
37         int clientPortion = dataCount / tasksCount;
38         Log.Print("Всего строк в матрице:{0}", dataCount);
39         Log.Print("Клиентов:{0}", tasksCount);
40         Log.Print("Кол-во строк на клиента:{0}", clientPortion);
41
42         for (int i = 0; i < tasksCount; i++) {
43             temp = new Task();
44             Log.Print("\nИнициализация счетчика умножаемых строк для
клиента #{0}", i + 1);
45             temp.start = i * clientPortion;
46             Log.Print("Начальная строка: {0}", temp.start + 1);
47             temp.stop = temp.start + clientPortion - 1;
48             Log.Print("Конечная строка: {0}", temp.stop + 1);
49             QueTasks.Enqueue(temp);           //добавление зада
чи в конец очереди
50         }
51         Log.Print("\nЗадачи успешно созданы и распределены!");
52     }
53
54     void CreateData() {
55         Log.Print("\n\nИсходные данные:");
56         /***ввод тестовых значений для проверки правильности вычислений
57         Console.Out.WriteLine("Матрица A:");
58         int k = 1;
59         for (int i = 0; i < m; i++) {
60             Console.Out.WriteLine();
61             for (int j = 0; j < n; j++) {

```

Ине. № дубл.	Подп. и дата	Взам. инв. №	Подп. и дата	Ине. № подл.	Ине. № инв.	Ли	Изм.	№ докум.	Подп.	Дат	Лист

```

62             A[i, j] = k;
63             k++;
64             Console.Out.Write("{0}\t", A[i, j].ToString());
65         }
66     }
67
68     Console.Out.WriteLine("\n\nВектор-столбец B:");
69     k = 1;
70     for (int j = 0; j < n; j++) {
71         B[j] = k;
72         k++;
73         Console.Out.Write("{0}\n", B[j].ToString());
74     }
75     /**
76
77
78     // заполнение случайными значениями
79     //Log.Print("Заполнение случайными значениями...\n
80     ");
81
82     //Random r = new Random();
83     //for (int i = 0; i < m; i++)
84     //{
85     //    for (int j = 0; j < n; j++)
86     //        A[i, j] = r.Next(100);
87     //}
88
89     //for (int j = 0; j < n; j++)
90     //    B[j] = r.Next(100);
91
92     //Log.Print("Исходная матрица и вектор-столбец успешно заполнены
93     случайными значениями!\n");
94     }
95
96     public void GetData(Task task, out int[] Btemp, out int[,] Atemp) {
97         Atemp = new int[m, n];
98         Btemp = new int[n];
99         Log.Print("\nКлиент начал получение данных для обработки!");
100        Console.Out.WriteLine("A[m,n] передаваемое клиенту:");
101        for (int i = task.start; i <= task.stop; i++) {
102            Console.Out.WriteLine();
103            for (int j = 0; j < n; j++) {
104                Atemp[i, j] = A[i, j];
105                Console.Out.Write("{0}\t", Atemp[i, j].ToString());

```

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

Ли	Изм.	№ докум.	Подп.	Дат



```

102         }
103     }
104     Console.Out.WriteLine("\n\nB[n] передаваемое клиенту:");
105     for (int j = 0; j < n; j++) {
106         Btemp[j] = B[j];
107         Console.Out.Write("{0}\n", Btemp[j].ToString());
108     }
109     Log.Print("Клиент получил данные для обработки!\n\n");
110 }
111
112 public Task GetTask() {
113     Log.Print("\nКлиент запросил задачу");
114     lock (tasksLock) {
115         if (QueTasks.Count == 0) { //если задачи кончились
116             Log.Print("Больше нет задач..."); //сообщим об этом
117             return null;
118         } else {
119             return QueTasks.Dequeue(); //если еще не кончились -
120             //вернем следующую задачу, извлеченную из очереди
121         }
122     }
123
124     public void Finish(int[] mas) {
125         lock (dataLock) {
126             for (int i = 0; i < n; i++) {
127                 C[i] += mas[i];
128             }
129             Log.Print("\nКлиент успешно завершил задачу!");
130         }
131
132         if (QueTasks.Count == 0) {
133             Console.Out.Write("\n\nПолученный результат:\n");
134             for (i = 0; i < m; i++) {
135                 Console.Out.WriteLine("C[{0}]: {1}", i + 1, C[i].ToString());
136             }
137         }
138     }
139 }
140
141 [Serializable]

```

Ине. № подл.	Подп. и дата
Ине. № дубл.	Взам. инв. №
Подп. и дата	
Ине. № подл.	

Ли	Изм.	№ докум.	Подп.	Дат
----	------	----------	-------	-----

```

142     public class Task {
143         public int start = 0, stop = 0;  //определение начала и конца диапазона
144     }
145
146     public class Log { //вывод записи в консоли на сервере
147     //     public static void Print(String msg, params int[] values) //использу
ется ключевое слово params для передачи неопределенного числа параметров в функцию
148     //     {
149     //         Console.WriteLine(msg, values);
150     //     }
151
152         //на случай проблем в работе params перегрузим метод Print
153         public static void Print(String msg) {
154             Console.WriteLine(msg);
155         }
156         public static void Print(String msg, int param1) {
157             Console.WriteLine(msg, param1);
158         }
159     }
160 }

```

### 3.3. Сервер

```

1 using System;
2
3 using System.Runtime.Remoting;
4 using System.Runtime.Remoting.Channels;
5 using System.Runtime.Remoting.Channels.Tcp;
6
7 using Lib;
8
9 namespace Server
10 {
11     class Server
12     {
13         TcpChannel channel;
14
15         public void Start() {
16             channel = new TcpChannel(8081);
17             ChannelServices.RegisterChannel(channel, false);
18             RemotingConfiguration.RegisterWellKnownServiceType(typeof(SharedObject),
данных", WellKnownObjectMode.Singleton);

```

Подп. и дата				
Взам. инв. №				
Инв. № дубл.				
Подп. и дата				
Инв. № подл.				
Ли	Изм.	№ докум.	Подп.	Дат

```

19         Log.Print("Сервер запущен");
20     }
21
22     public void Stop() {
23         ChannelServices.UnregisterChannel(channel);
24         Log.Print("Сервер остановлен");
25     }
26
27     static void Main(string[] args) {
28         Server srv = new Server();
29         srv.Start();
30         Console.In.ReadLine();
31         srv.Stop();
32     }
33 }
34 }

```

Инв. № подл	Подп. и дата				Лист
	Взам. инв. №				
	Инв. № дубл.				
Инв. № подл	Подп. и дата				10
	Взам. инв. №				
	Инв. № дубл.				
Ли	Изм.	№ докум.	Подп.	Дат	Лабораторная работа № 2

4. Результат выполнения

4.1. Клиент

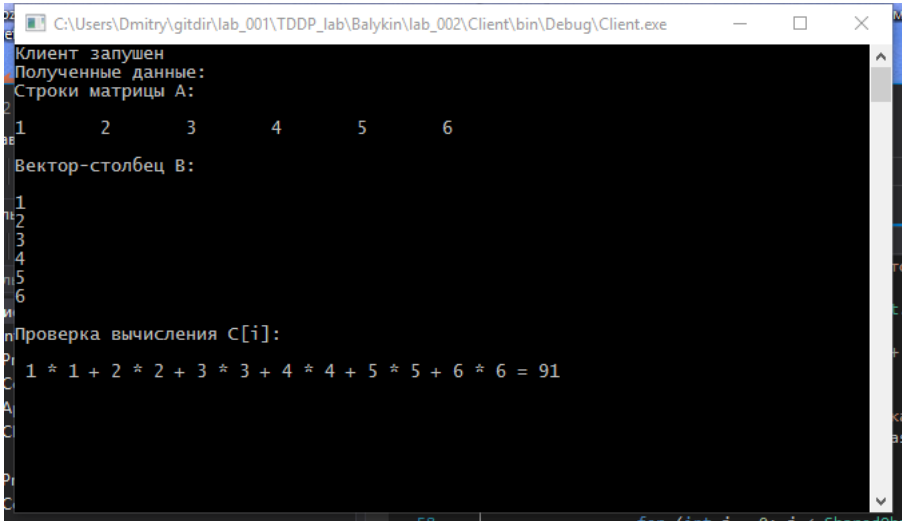


Рисунок 4.1. Пример работы клиента.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	Подп. и дата	Лабораторная работа № 2				Лист
										8
Ли	Изм.	№ докум.	Подп.	Дат						

## 4.2. Сервер

```

cmd Командная строка - Server
eg
Сервер запущен

Исходные данные:
Матрица A:
1      2      3      4      5      6
7      8      9      10     11     12
13     14     15     16     17     18
19     20     21     22     23     24
25     26     27     28     29     30
31     32     33     34     35     36

Вектор-столбец B:
1
2
3
4
5
6

Создание задач...

Всего строк в матрице:6
Клиентов:6
Кол-во строк на клиента:1

Инициализация счетчика умножаемых строк для клиента #1
Начальная строка: 1
Конечная строка: 1

Инициализация счетчика умножаемых строк для клиента #2
Начальная строка: 2
Конечная строка: 2

Инициализация счетчика умножаемых строк для клиента #3
Начальная строка: 3
Конечная строка: 3

Инициализация счетчика умножаемых строк для клиента #4
Начальная строка: 4
Конечная строка: 4

Инициализация счетчика умножаемых строк для клиента #5
Начальная строка: 5
Конечная строка: 5

Инициализация счетчика умножаемых строк для клиента #6
Начальная строка: 6
Конечная строка: 6

Задачи успешно созданы и распределены!

Клиент запросил задачу

Клиент начал получение данных для обработки!
A[m,n] передаваемое клиенту:
1      2      3      4      5      6

B[n] передаваемое клиенту:
1
2
3
4
5
6

Клиент получил данные для обработки!

```

Рисунок 4.2. Пример работы сервера.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						
					Ли	Изм.	№ докум.	Подп.	Дат	Лист
					Лабораторная работа № 2					9

```

cmd Командная строка - Server

Клиент запросил задачу
Клиент начал получение данных для обработки!
A[m,n] передаваемое клиенту:
19      20      21      22      23      24
B[n] передаваемое клиенту:
1
2
3
4
5
6
Клиент получил данные для обработки!

Клиент успешно завершил задачу!
Клиент запросил задачу
Клиент начал получение данных для обработки!
A[m,n] передаваемое клиенту:
25      26      27      28      29      30
B[n] передаваемое клиенту:
1
2
3
4
5
6
Клиент получил данные для обработки!

Клиент успешно завершил задачу!
Клиент запросил задачу
Клиент начал получение данных для обработки!
A[m,n] передаваемое клиенту:
31      32      33      34      35      36
B[n] передаваемое клиенту:
1
2
3
4
5
6
Клиент получил данные для обработки!

Клиент успешно завершил задачу!

Полученный результат:
C[1]: 91
C[2]: 217
C[3]: 343
C[4]: 469
C[5]: 595
C[6]: 721
Клиент запросил задачу
Больше нет задач...

```

Рисунок 4.3. Пример работы сервера в конце.

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

Ли	Изм.	№ докум.	Подп.	Дат

Вывод

В результате выполнения лабораторной работы был получен программный комплекс, состоящий из сервера и клиента и реализующий алгоритм умножения матрицы на вектор-столбец.

Инв. № подл.	Подп. и дата				Инв. № дубл.	Взам. инв. №	Подп. и дата	
Ли	Изм.	№ докум.	Подп.	Дат	Лабораторная работа № 2			Лист
								11