

Правительство Российской Федерации

**Федеральное государственное автономное образовательное
учреждение высшего образования «Национальный исследовательский
университет «Высшая школа экономики»**

Факультет компьютерных наук
Департамент программной инженерии

**Отчет к домашнему заданию
По дисциплине
«Архитектура вычислительных систем»**

Работу выполнил:
Студент группы БПИ-194 Горбач И.В.
Вариант 7

Москва 2020

Задание

7. Военная задача. Анчуария и Тарантерия – два крохотных латиноамериканских государства, затерянных в южных Андах. Диктатор Анчуарии, дон Федерико, объявил войну диктатору Тарантерии, дону Эрнандо. У обоих диктаторов очень мало солдат, но очень много снарядов для минометов, привезенных с последней американской гуманитарной помощью. Поэтому армии обеих сторон просто обстреливают наугад территорию противника, надеясь поразить что-нибудь ценное. Стрельба ведется по очереди до тех пор, пока либо не будут уничтожены все цели, либо стоимость потраченных снарядов не превысит суммарную стоимость всего того, что ими можно уничтожить. Создать многопоточное приложение, моделирующее военные действия. **Модель**

Итеративный параллелизм используется для реализации нескольких потоков (часто идентичных), каждый из которых содержит циклы. Потоки программы, описываются итеративными функциями и работают совместно над решением одной задачи.

Решение

Для реализации данной задачи была использовано 2 библиотеки: threads и mutex

Ввод: входных данных нет, можно в самом коде менять площадь территории городов, стоимость одного снаряда.

```
using namespace std;

mutex mtx;
const unsigned int dim_1=3; //количество клеток по вертикали
const unsigned int dim_2=3; //количество клеток по горизонтали

///
/// \param min нижнее значение
/// \param max верхнее значение
/// \return рандомное число между мин и макс
int getRandomNumber(int min, int max)
{
    static const double fraction = 1.0 / (static_cast<double>(RAND_MAX) + 1.0);
    // Равномерно распределяем рандомное число в нашем диапазоне
    return static_cast<int>(rand() * fraction * (max - min + 1) + min);
}
```

Площадь территории заполняется рандомайзером

КЛАСС CITY

Поля в классе City:

```
class City{
public:
    string name;//Название города
```

```

string name_of_dictator; //имя диктатор
int territory[dim_1][dim_2]; //территория города - матрица по дефолту 10
на 10
int cost_summ=0; //общая сумма клеток города
//конструктор класса с параметром имени диктатора

```

Методы класса city:

```

City(string city_name, string dict_name); //инициализатор конструктора
~City(); //иниц деструктора

void Create_Territory() {
    for (auto & i : territory) {
        for (int & j : i) {
            int cost=getRandomNumber(0,100);
            j=cost; // rand mod 100 for creating normal values
            cost_summ+=cost; //и в сумму тоже добавляем ранд
        }
    }
    Print_Territory();
}

void Print_Territory() {
    cout<< "territory: "<< this->name<<endl << "Current cost of
territory= "<< cost_summ << endl;
    for (auto & i : territory) {
        for (int j : i) {
            cout << j << " ";
        }
        cout << endl;
    }
}
};

```

Метод Create_Territory отвечает за инициализацию поля territory-что является территорией города. Метод Print_Territory отвечает за вывод на экран территории города.

КОНСТРУТОР И ДЕКТРУКТОР

```

///
/// \param city_name название города
/// \param dict_name имя диктатора
City::City(string city_name, string dict_name) {
    this->name_of_dictator=dict_name;
    this->name=city_name;
    this->Create_Territory();
}

City::~~City() {
}

```

МЕТОД MakeMove()

Этот метод, метод совершения выстрела по другому городу

```

void MakeMove(City& other, int first_rand,int second_rand,thread& thread){
    mtx.lock();//блокируем мьютекс
    cout <<"-----" << endl;
    cout <<"id потоковой пушки, стреляющей в : "<<other.name<<" равен:
"<<thread.get_id()<<endl;//выводим значение текущего потока
    mtx.unlock();//разблокируем мьютекс
    mtx.lock();//блокируем мьютекс
    cout<<other.name << " под огнем! Атакована территория: "<<first_rand+1<<"
"<<second_rand+1 << endl;//вывод атакованного города
    if (other.territory[first_rand][second_rand]==0){
        cout << "Здесь было пусто, возможно сюда уже стреляли"<< endl;
    }
    else {
        cout << "Территория была обстрелена, стоимость ущерба составила:
"<<other.territory[first_rand][second_rand]<<endl;
        other.cost_summ-=other.territory[first_rand][second_rand];
        other.territory[first_rand][second_rand] = 0;
    }
    this thread::sleep_for(chrono::milliseconds(200));//засыпаем на 200
миллисекунд
    other.Print_Territory();
    cout <<"Охлаждаем пушки!" << endl;
    mtx.unlock();//разблокируем мьютекс
}

```

Сначала мьютекс закрывается, чтобы правильно вывести разделительные полосы и номер текущего выполняемого потока, потом мьютекс разблокируется и блокируется заново, происходит обстрел вражеской территории, обстреливаемая территория выбирается случайно, если раньше значение стоимости вражеской территории было равно 0, то выводится: «Здесь было пусто, возможно сюда уже стреляли», если раньше там было что то, то выводится «"Территория была обстрелена, стоимость ущерба составила: "<<other.territory[first_rand][second_rand]», из общей стоимости всей территории вычитается стоимость разбомбленной территории и стоимость разбомбленной территории становится равно 0. Затем текущий поток отправляется спать на 200 миллисекунд, имитируя перезарядку, происходит вывод обстрелянного города в консоль и мьютекс разблокируется.

МЕТОД main:

```

int main() {
    cout<<"MAIN TREAD ID:"<<this_thread::get_id()<<endl;//номер мэйн
потока
    system("chcp 65001");
    int bullet_cost = 5;//стоимость 1 пули
    City Anchyuria = *new City("Anchyuria","Don Federico");//создание
анчурии
    City Taranteria = *new City("Taranteria","Don Hernando");//создание
тарантерии
    //создание треда анчурии и треда тарантерии, на основе MakeMove
    while (Anchyuria.cost_summ > bullet_cost&&Anchyuria.cost_summ>0&&
Taranteria.cost_summ>0 && Taranteria.cost_summ > bullet_cost) {

```

```

        thread anch(MakeMove, ref(Taranteria), getRandomNumber(0, dim_1-1), getRandomNumber(0, dim_2-1), ref(anch));
        thread taran(MakeMove, ref(Anchyuria), getRandomNumber(0, dim_1-1), getRandomNumber(0, dim_2-1), ref(taran));
        anch.join();
        taran.join();
    }
    if (Anchyuria.cost_summ<0||Anchyuria.cost_summ<bullet_cost){
        cout<<"В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ
ПРОИГРАЛА:"<<Anchyuria.name<<endl;
    }
    else if (Taranteria.cost_summ<0||Taranteria.cost_summ<bullet_cost){
        cout<<"В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ
ПРОИГРАЛА:"<<Taranteria.name<<endl;
    }
    system("exit");
    return 0;
}

```

Сначала в консоль выводится айди основного потока программы, затем ставится русская локаль. Создается переменная для стоимости одного патрона. Создается 2 экземпляра класса City, цикл while выполняется пока стоимость территории больше чем 0 или чем стоимость 1 пули. В теле цикла создается 2 потока(костыль, не знал как сделать иначе), один поток имитирует обстрел Тарантерии Анчурией, другой поток имитирует обстрел Анчурии Тарантерией. Параметрами являются обстреливаемые города, два интовых значения, отвечающие за строку и столбец в матрице вражеской территории и поток, для вывода id текущего потока. Затем у обоих потоков вызывается метод join(). После выхода из тела цикла, осуществляется проверка на проигравшего, и завершение программы.

Код программы

```

#include <iostream>
#include "string"
#include <thread>
#include <mutex>

/*
 * Горбач Илья Вадимович
 * БПИ 194
 * Вариант №7
 * 7. Военная задача. Анчуария и Тарантерия – два крохотных
латиноамериканских государства, затерянных в южных Андах. Диктатор
Анчуарии, дон Федерико, объявил войну диктатору Тарантерии, дону
Эрнандо. У обоих диктаторов очень мало солдат, но очень много снарядов
для минометов, привезенных с последней американской гуманитарной
помощью. Поэтому армии обеих сторон просто обстреливают наугад
территорию противника, надеясь поразить что-нибудь ценное. Стрельба
ведется по очереди до тех пор, пока либо не будут уничтожены все цели,
либо стоимость потраченных снарядов не превысит суммарную стоимость
всего того, что ими можно уничтожить. Создать многопоточное приложение,
моделирующее военные действия.
 */

```

```

using namespace std;

mutex mtx;
const unsigned int dim_1=3; //количество клеток по вертикали
const unsigned int dim_2=3; //количество клеток по горизонтали

///
/// \param min нижнее значение
/// \param max верхнее значение
/// \return случайное число между мин и макс
int getRandomNumber(int min, int max)
{
    static const double fraction = 1.0 / (static_cast<double>(RAND_MAX) +
1.0);
    // Равномерно распределяем случайное число в нашем диапазоне
    return static_cast<int>(rand() * fraction * (max - min + 1) + min);
}

//Класс город
class City{
public:
    string name;//Название города
    string name_of_dictator; //имя диктатор
    int territory[dim_1][dim_2]; //территория города - матрица по умолчанию 10
на 10
    int cost_summ=0; //общая сумма клеток города
    //конструктор класса с параметром имени диктатора
    City(string city_name,string dict_name); //инициализатор конструктора
    ~City(); //иниц деструктора

    void Create_Territory(){
        for (auto & i : territory) {
            for (int & j : i) {
                int cost=getRandomNumber(0,100);
                j=cost; // rand mod 100 for creating normal values
                cost_summ+=cost; //и в сумму тоже добавляем ранд
            }
        }
        Print_Territory();
    }

    void Print_Territory(){
        cout<< "territory: "<< this->name<<endl << "Current cost of
territory= "<< cost_summ << endl;
        for (auto & i : territory) {
            for (int j : i) {
                cout << j << " ";
            }
            cout << endl;
        }
    }
};

///
/// \param city_name название города
/// \param dict_name имя диктатора
City::City(string city_name, string dict_name) {
    this->name_of_dictator=dict_name;
    this->name=city_name;
    this->Create_Territory();
}

City::~City() {
}

void MakeMove(City& other, int first_rand,int second_rand,thread& thread){
    mtx.lock(); //блокируем мьютекс
    cout << "-----

```

```

-----" << endl;
    cout <<"id потоковой пушки, стреляющей в : "<<other.name<<" равен:
"<<thread.get_id()<<endl;//выводим значение текущего потока
    mtx.unlock();//разблокируем мьютекс
    mtx.lock();//блокируем мьютекс
    cout<<other.name << " под огнем! Атакована территория: "<<first_rand+1<<"
"<<second_rand+1 << endl;//вывод атакованного города
    if (other.territory[first_rand][second_rand]==0){
        cout << "Здесь было пусто, возможно сюда уже стреляли"<< endl;
    }
    else {
        cout << "Территория была обстрелена, стоимость ущерба составила:
"<<other.territory[first_rand][second_rand]<<endl;
        other.cost_summ+=other.territory[first_rand][second_rand];
        other.territory[first_rand][second_rand] = 0;
    }
    this_thread::sleep_for(chrono::milliseconds(200));//засыпаем на 200
миллисекунд
    other.Print_Territory();
    cout <<"Охлаждаем пушки!" << endl;
    mtx.unlock();//разблокируем мьютекс
}

int main() {
    cout<<"MAIN TREAD ID:"<<this_thread::get_id()<<endl;//номер мейн
потока
    system("chcp 65001");
    int bullet_cost = 5;//стоимость 1 пули
    City Anchuria = *new City("Anchuria","Don Federico");//создание
анчурии
    City Taranteria = *new City("Taranteria","Don Hernando");//создание
тарантерии
    //создание треда анчурии и треда тарантерии, на основе MakeMove
    while (Anchuria.cost_summ > bullet_cost&&Anchuria.cost_summ>0&&
Taranteria.cost_summ>0 && Taranteria.cost_summ > bullet_cost) {
        thread anch(MakeMove,ref(Taranteria),getRandomNumber(0,dim_1-
1),getRandomNumber(0,dim_2-1),ref(anch));
        thread taran(MakeMove, ref(Anchuria),getRandomNumber(0,dim_1-
1),getRandomNumber(0,dim_2-1),ref(taran));
        anch.join();
        taran.join();
    }
    if (Anchuria.cost_summ<0||Anchuria.cost_summ<bullet_cost){
        cout<<"В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ
ПРОИГРАЛА:"<<Anchuria.name<<endl;
    }
    else if (Taranteria.cost_summ<0||Taranteria.cost_summ<bullet_cost){
        cout<<"В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ
ПРОИГРАЛА:"<<Taranteria.name<<endl;
    }
    system("exit");
    return 0;
}

```

Тестирование

Пример 1:

```
Run: MICRO_PROJECT_2
C:\Clion\MICRO_PROJECT_2\cmake-build-debug\MICRO_PROJECT_2.exe
MAIN TREAD ID:1
Active code page: 65001
territory: Anchyuria
Current cost of territory= 471
0 56 19
81 59 48
35 98 83
territory: Taranteria
Current cost of territory= 376
75 17 86
71 51 38
1 9 36

-----
id поточковой пушки, стреляющей в : Taranteria равен: 2
Taranteria под огнем! Атакована территория: 1 1
Территория была обстрелена, стоимость ущерба составила: 75
territory: Taranteria
Current cost of territory= 301
0 17 86
71 51 38
1 9 36
Охлаждаем пушки!

-----
id поточковой пушки, стреляющей в : Anchyuria равен: 3
Anchyuria под огнем! Атакована территория: 2 2
Территория была обстрелена, стоимость ущерба составила: 59
territory: Anchyuria
Current cost of territory= 283
0 56 19
0 0 0
35 98 83
Охлаждаем пушки!
```

```
Run: MICRO_PROJECT_2
-----
id поточковой пушки, стреляющей в : Anchyuria равен: 7
Anchyuria под огнем! Атакована территория: 2 2
Территория была обстрелена, стоимость ущерба составила: 59
territory: Anchyuria
Current cost of territory= 283
0 56 19
0 0 0
35 98 83
Охлаждаем пушки!

-----
id поточковой пушки, стреляющей в : Anchyuria равен: 9
Anchyuria под огнем! Атакована территория: 2 2
Здесь было пусто, возможно сюда уже стреляли
territory: Anchyuria
Current cost of territory= 283
0 56 19
0 0 0
35 98 83
Охлаждаем пушки!

-----
id поточковой пушки, стреляющей в : Taranteria равен: 8
Taranteria под огнем! Атакована территория: 2 1
Территория была обстрелена, стоимость ущерба составила: 71
territory: Taranteria
Current cost of territory= 301
0 17 86
71 51 38
1 9 36
Охлаждаем пушки!
```


The image displays two screenshots of a C++ IDE (Visual Studio Code) showing the execution of a program named MICRO_PROJECT_2. The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help), a toolbar with icons for running, debugging, and other actions, and a sidebar with project and file explorer views.

The first screenshot shows the output of the program in the Run window. The output is divided into three sections by dashed lines, each representing a different state of the program. The first section shows the program attacking the territory of Anchyuria, with a current cost of 283. The second section shows the program attacking the territory of Anchyuria again, with a current cost of 283. The third section shows the program attacking the territory of Taranteria, with a current cost of 71.

The second screenshot shows the output of the program in the Run window, continuing from the first screenshot. The output is divided into three sections by dashed lines, each representing a different state of the program. The first section shows the program attacking the territory of Taranteria, with a current cost of 126. The second section shows the program attacking the territory of Anchyuria, with a current cost of 125. The third section shows the program attacking the territory of Taranteria, with a current cost of 40.

Конец примера 1:

```
CMakeLists.txt x main.cpp x thread x
Run: MICRO_PROJECT_2 x
Охлаждаем пушки!
-----
id потоковой пушки, стреляющей в : Taranteria равен: 26
Taranteria под огнем! Атакована территория: 1 1
Здесь было пусто, возможно сюда уже стреляли
territory: Taranteria
Current cost of territory= 40
0 0 0
0 0 30
1 9 0
Охлаждаем пушки!
-----
id потоковой пушки, стреляющей в : Anchyuria равен: 27
Anchyuria под огнем! Атакована территория: 3 2
Территория была обстрелена, стоимость ущерба составила: 90
territory: Anchyuria
Current cost of territory= 0
0 0 0
0 0 0
0 0 0
Охлаждаем пушки!
В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ ПРОИГРАЛА:Anchyuria

Process finished with exit code 0
```

Пример был выполнен для размера территории 3:3

Пример 2 для территории 5 на 5:

Начало:

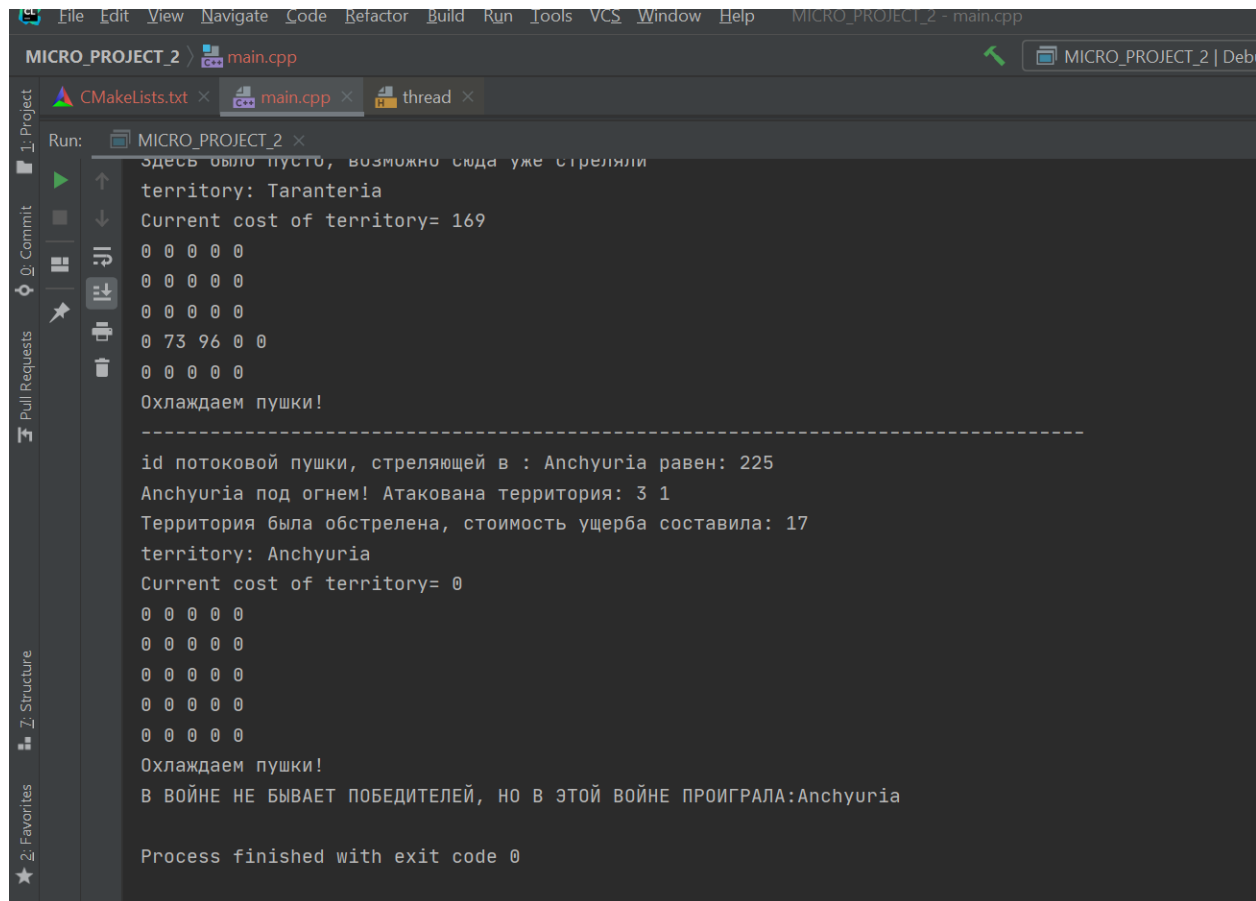
```
Run: MICRO_PROJECT_2 x
C:\CLion\MICRO_PROJECT_2\cmake-build-debug\MICRO_PROJECT_2.exe
MAIN TREAD ID:1
Active code page: 65001
territory: Anchyuria
Current cost of territory= 1033
0 56 19 81 59
48 35 90 83 75
17 86 71 51 30
1 9 36 14 16
99 45 12 0 0
territory: Taranteria
Current cost of territory= 1334
38 53 57 60 61
16 66 45 35 5
61 79 81 52 30
88 73 96 93 54
14 46 23 87 21
-----
id потоковой пушки, стреляющей в : Taranteria равен: 2
Taranteria под огнем! Атакована территория: 5 4
Территория была обстрелена, стоимость ущерба составила: 87
territory: Taranteria
Current cost of territory= 1247
38 53 57 60 61
```

Выполнение идет довольно долго, в связи с засыпаниями на 200 миллисекунд и попытками попасть рандомайзером в ранее необстрелянную клетку города

Середина программы:

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help MICRO_PROJECT_2 - main.cpp
MICRO_PROJECT_2 x main.cpp
CMakeLists.txt x main.cpp x thread x
Run: MICRO_PROJECT_2 x
-----
id потоковой пушки, стреляющей в : Taranteria равен: 100
Taranteria под огнем! Атакована территория: 4 4
Здесь было пусто, возможно сюда уже стреляли
territory: Taranteria
Current cost of territory= 199
0 0 0 0 0
0 0 0 0 0
0 0 0 0 30
0 73 96 0 0
0 0 0 0 0
Охлаждаем пушки!
-----
id потоковой пушки, стреляющей в : Anchyuria равен: 101
Anchyuria под огнем! Атакована территория: 2 3
Здесь было пусто, возможно сюда уже стреляли
territory: Anchyuria
Current cost of territory= 219
0 0 0 0 0
0 0 0 0 0
17 86 71 0 0
0 0 0 0 0
0 45 0 0 0
Охлаждаем пушки!
-----
```

Конец программы:



```
Run: MICRO_PROJECT_2 x
Здесь было пусто, возможно сюда уже стреляли
territory: Taranteria
Current cost of territory= 169
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 73 96 0 0
0 0 0 0 0
Охлаждаем пушки!
-----
id потоковой пушки, стреляющей в : Anchyuria равен: 225
Anchyuria под огнем! Атакована территория: 3 1
Территория была обстрелена, стоимость ущерба составила: 17
territory: Anchyuria
Current cost of territory= 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Охлаждаем пушки!
В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ ПРОИГРАЛА:Anchyuria

Process finished with exit code 0
```

Список используемых источников

1. <https://plugins.jetbrains.com/plugin/8352-c-c--single-file-execution>
2. <https://coderoad.ru/32030935/>
3. <https://ravesli.com/urok-71-generatsiya-sluchajnyh-chisel-funktsii-srand-i-rand/#toc-1>
4. <https://www.cyberforum.ru/cpp-beginners/thread956351.html>
5. <https://www.youtube.com/watch?v=DFgtADi0Mjo>
6. <http://www.softcraft.ru/>