

Правительство Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования «Национальный

исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Департамент программной инженерии

Отчет к микропроекту по дисциплине

«Архитектура вычислительных систем»

Работу выполнил:

Студент группы БПИ-196 Мосолков Е.Н.

Москва 2020

Содержание

1. ЗАДАЧА.....	2
2. РЕШЕНИЕ.....	3
3. КОД ПРОГРАММЫ	4
4. ТЕСТИРОВАНИЕ.....	6
5. ПРИЛОЖЕНИЕ	7

1. ЗАДАЧА

16. *Задача о клумбе.* На клумбе растет 40 цветов, за ними непрерывно следят два садовника и поливают увядшие цветы, при этом оба садовника очень боятся полить один и тот же цветок. Создать многопоточное приложение, моделирующее состояния клумбы и действия садовников. Для изменения состояния цветов создать отдельный поток.

2. РЕШЕНИЕ

Создаем потоки садовников (1 и 2) и поток менеджера сада

Поток садовника – поливает клумбу, если та не полита

Поток менеджера сада – изменяет состояние политой клумбы и делает ее неполитой

Клумбы представляют из себя массив boolean значений, которые означают, что клумба полита либо же не полита.

После выполнения задачи садовник засыпает на 500 мс, а менеджер сада на 1000 мс (так задача может завершиться)

В программе реализовано 3 функции:

`changeFlowerBeds` – меняет статус случайной клумбы, делая ее неполитой

`findUnfilledFlowerBeds` – находит какие клумбы не политы и поливает клумбу или пишет в консоль, что все клумбы политы

`water` – поливает нужную клумбу и выводит сообщение об этом в консоль

3. КОД ПРОГРАММЫ

```

#include <string>
#include <iostream>
#include <mutex>

using namespace std;

mutex mtx;
static bool flowerBeds[40];
const int gardenerSleepTime = 500;
const int gardenManagerSleepTime = 1000;

void changeFlowerBeds(int count);

void water(int i, int thread);

void findUnfilledFlowerBed(int unfilledFlowerBeds, int thread);

int main(int argc, char** argv) {
    if (argc != 2) {
        cout << "Invalid number of arguments given";
        return -1;
    }
    int actions = stoi(argv[1]);
    if (actions <= 0) {
        cout << "Invalid number of actions";
        return -1;
    }
    // This thread changes some flower beds status to unfilled
    thread gardenManager(changeFlowerBeds, actions * 2);
    // Threads which checks flowered beds and water flowers if flower bed is unfilled
    thread firstGardener(findUnfilledFlowerBed, actions, 1);
    thread secondGardener(findUnfilledFlowerBed, actions, 2);
    gardenManager.join();
    firstGardener.join();
    secondGardener.join();
    return 0;
}

/// This function change status of specific count of flower beds to unfilled.
/// Works in parallel with gardeners
/// \param count - count of flower beds to change
void changeFlowerBeds(int count) {
    for (int j = 0; j < count; ++j) {
        this_thread::sleep_for(chrono::milliseconds(gardenManagerSleepTime));
        int i = rand() % 40;
        flowerBeds[i] = false;
        cout << to_string(i) + " flower bed is not watered\n";
    }
}

/// This function distributes tasks among gardeners
/// \param unfilledFlowerBeds - the amount of flower beds which are not filled
/// \param thread - thread id
void findUnfilledFlowerBed(int unfilledFlowerBeds, int thread) {
    for (int j = 0; j < unfilledFlowerBeds; j++ ) {

```

```

        this_thread::sleep_for(chrono::milliseconds(gardenerSleepTime));
        mtx.lock();
        for (int i = 0; i < 40; ++i) {
            if (!flowerBeds[i]) {
                water(i, thread);
                break;
            }
            if (i == 39) {
                cout << "All flower beds are filled\n";
            }
        }
        mtx.unlock();
    }
}

/// This function water specific flower bed
/// \param i - index of flower bed to water
/// \param thread - thread id
void water(int i, int thread) {
    if (flowerBeds[i]) return;
    this_thread::sleep_for(chrono::milliseconds(gardenerSleepTime));
    flowerBeds[i] = true;
    cout << to_string(i + 1) + " was watered by " + to_string(thread) + "
gardener\n";
}

```

4. ТЕСТИРОВАНИЕ

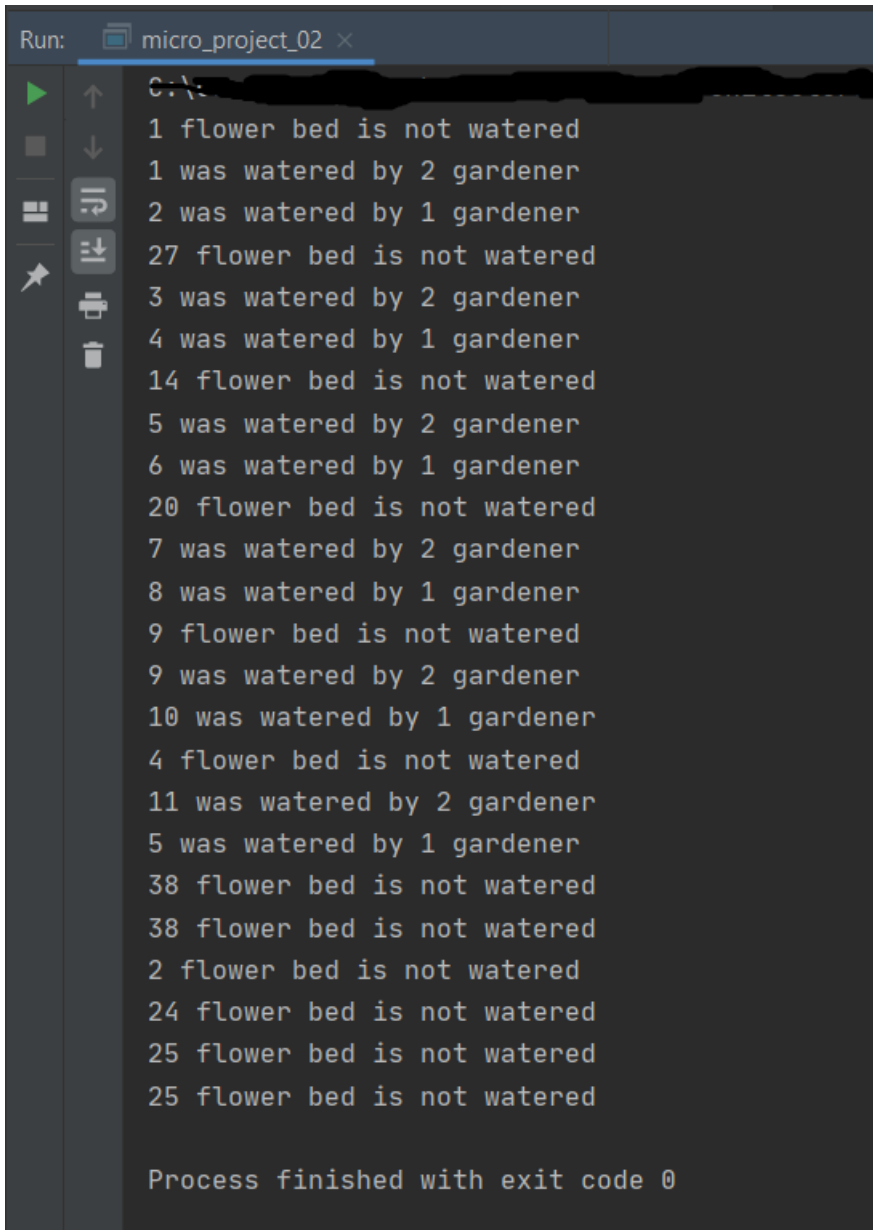
4.1. Параметры командной строки

При вводе некорректного числа аргументов, программа завершается с кодом -1, и выводит сообщение в консоль о том, что количество аргументов командной строки недопустимо. При вводе некорретного фргумента программа также завершиться с кодом -1 и выведет сообщение о том что аргумент некорректный

4.2. Корректные данные

При тестировании запускаем программу с разными аргументами. Во всех случаях программа работает корректно (см. приложение)

5. ПРИЛОЖЕНИЕ



```
Run: micro_project_02 x
C:\...
1 flower bed is not watered
1 was watered by 2 gardener
2 was watered by 1 gardener
27 flower bed is not watered
3 was watered by 2 gardener
4 was watered by 1 gardener
14 flower bed is not watered
5 was watered by 2 gardener
6 was watered by 1 gardener
20 flower bed is not watered
7 was watered by 2 gardener
8 was watered by 1 gardener
9 flower bed is not watered
9 was watered by 2 gardener
10 was watered by 1 gardener
4 flower bed is not watered
11 was watered by 2 gardener
5 was watered by 1 gardener
38 flower bed is not watered
38 flower bed is not watered
2 flower bed is not watered
24 flower bed is not watered
25 flower bed is not watered
25 flower bed is not watered

Process finished with exit code 0
```



```
Run: micro_project_02 x
15 was watered by 1 gardener
2 flower bed is not watered
16 was watered by 2 gardener
3 was watered by 1 gardener
24 flower bed is not watered
17 was watered by 2 gardener
18 was watered by 1 gardener
25 flower bed is not watered
19 was watered by 2 gardener
20 was watered by 1 gardener
25 flower bed is not watered
21 was watered by 2 gardener
22 was watered by 1 gardener
1 flower bed is not watered
23 was watered by 2 gardener
2 was watered by 1 gardener
27 flower bed is not watered
1 flower bed is not watered
11 flower bed is not watered
35 flower bed is not watered
22 flower bed is not watered
27 flower bed is not watered
36 flower bed is not watered
31 flower bed is not watered
4 flower bed is not watered
22 flower bed is not watered
33 flower bed is not watered
12 flower bed is not watered
22 flower bed is not watered

Process finished with exit code 0
```