

Задача: 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, который еще не начал вянуть. Создать многопоточное приложение, моделирующее состояния цветков на клумбе и действия садовников. Для изменения состояния цветов создать отдельный поток (а не 40 потоков для каждого цветка), который может задавать одновременное начало увядания для нескольких цветков.

Входные данные:

Последовательность нулей и единиц, 0 - завявший цветок, 1 цветущий, в общем числе 40.

Ввод возможен внутри консоли, через командную строку, а также через файл (путь до файла указывается через консоль или командную строку), доступна случайная генерация.

Потоки:

Главный поток - поток цветов.

Взаимодействующие равные дочерние потоки - садовники, которые поливают цветы (меняют 0 на 1).

Алгоритм : Вначале происходит инициализация массива, представляющего цветы, дополнительно к этому создается вектор, в который складываются индексы увядающих цветов. Цветы могут сами вянуть с определенной вероятностью (например 20%, как в коде ниже), садовники случайным образом достают индекс из вектора и поливают (меняют 0 на 1) цветок с этим индексом из массива цветов. Для отслеживания в консоль пишется массив цветов, когда увядают случайные цветки, и когда поливается какой-то цветок, если нет завявших цветов, то в консоль пишется сообщение "All flowers are beautiful!" Цветы и садовники - это 3 разных потока.

Пример ввода:

В консоли:

```
random  
direct 1 0 1 0 1 0 1 1 1 ...
```

file <путь до файла>

В командной строке:

./main random

./main direct 1 0 1 0 1 0 1 1 1 ...

./main file <путь до файла>

Код C++:

```
#include <iostream>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <pthread.h>
#include <ctime>
#include <vector>
#include <cstring>
#include <string>
#include <fstream>

//Массив цветов
const int arrSize = 40;
int Flowers[arrSize];
//Вектор индексов увядающих цветов
std::vector<int> FadingFlowers;
//двоичный семафор
pthread_mutex_t mutex;
//Увядание
void *Fading(void *param) {
    srand(time(NULL));
    while (true) {
        std::string sf;
        pthread_mutex_lock(&mutex); //протокол входа в КС:
        закрыть двоичный семафор
        for (int i = 0; i < arrSize; i++) {
            if (Flowers[i] == 1) {
                bool to_fade_or_not_to_fade = rand() % 5
                == 0;
                if (to_fade_or_not_to_fade) {
                    Flowers[i] = 0;
                }
            }
        }
        pthread_mutex_unlock(&mutex);
    }
}
```

```

        FadingFlowers.push_back(i);
    }
}
sf += std::to_string(Flowers[i]);
}
std::cout << "time = " << clock() << " | Flowers:
" << sf << "\n";
pthread_mutex_unlock(&mutex); //протокол выхода из
КС:
    sleep(4);
}
return nullptr;
}
//Полив
void *Watering(void *param) {
    srand(time(NULL));
    int MyNum = *((int *) param);
    while (true) {
        pthread_mutex_lock(&mutex); //протокол входа в КС:
закреть двоичный семафор
        if (FadingFlowers.size() > 0) {
            int NumberOfMyFlower = rand() %
FadingFlowers.size();
            int MyFlower =
FadingFlowers[NumberOfMyFlower];
            FadingFlowers.erase(FadingFlowers.begin() +
NumberOfMyFlower);
            if (Flowers[MyFlower] == 1) {
                std::cout << "Error!!!";
            } else {
                Flowers[MyFlower] = 1;
            }
            std::cout << "time = " << clock() << " |
Gardener #" << MyNum + 1 << " watered flower #" <<
MyFlower + 1
                << "\n";
        } else {
            std::cout << "All flowers are beautiful\n";
        }
    }
}

```

```

        pthread_mutex_unlock(&mutex); //протокол выхода из
КС:
        sleep(1);
    }
    return nullptr;
}

int main(int argc, char *argv[]) {
    srand(time(NULL));
    //инициализация двоичного семафора
    pthread_mutex_init(&mutex, NULL);
    if (argc != 1) {
        if (strcmp(argv[1], "random") == 0) {
            for (int i = 0; i < arrSize; i++) {
                Flowers[i] = rand() % 2;
                if (Flowers[i] == 0) {
                    FadingFlowers.push_back(i);
                }
            }
        } else if (strcmp(argv[1], "direct") == 0) {
            for (int i = 0; i < argc - 2 && i < arrSize;
i++) {

                Flowers[i] = std::stoi(argv[i + 2]);
                if (Flowers[i] != 0 && Flowers[i] != 1) {
                    std::cout << "Input Error";
                    return -1;
                }
            }
            for (int i = 0; i < arrSize; i++) {
                if (Flowers[i] == 0) {
                    FadingFlowers.push_back(i);
                }
            }
        } else if (strcmp(argv[1], "file") == 0) {
            if (argc < 3) {
                std::cout << "Input Error";
                return -1;
            }
        }
    }
}

```

```

std::ifstream in(argv[2]);
if (in.is_open()) {
    int counter = 0;
    std::string x;
    while (in >> x && counter < 40) {
        if (x != "0" && x != "1") {
            std::cout << "Input Error";
            return -1;
        }
        Flowers[counter] = std::stoi(x);
        counter++;
    }
    for (int i = 0; i < arrSize; i++) {
        if (Flowers[i] == 0) {
            FadingFlowers.push_back(i);
        }
    }
} else {
    std::cout << "Input Error";
    return -1;
}
in.close();
} else {
    std::cout << "Input Error";
    return -1;
}
} else {
    std::cout << "Enter \"random\", \"direct\" or
\"file\": ";
    std::string str;
    while (true) {
        std::cin >> str;
        if (str == "random" || str == "direct" || str
== "file") {
            break;
        }
    }
    if (str == "random") {
        for (int i = 0; i < arrSize; i++) {

```

```

        Flowers[i] = rand() % 2;
        if (Flowers[i] == 0) {
            FadingFlowers.push_back(i);
        }
    }
} else if (str == "direct") {
    for (int i = 0; i < arrSize; i++) {
        std::cout << "Enter Flower[" << i << "], 0
- fading, 1 - beautiful: ";
        while (true) {
            int x;
            std::cin >> x;
            if (x == 0 || x == 1) {
                Flowers[i] = x;
                break;
            }
        }
        for (int i = 0; i < arrSize; i++) {
            Flowers[i] = rand() % 2;
            if (Flowers[i] == 0) {
                FadingFlowers.push_back(i);
            }
        }
    }
} else if (str == "file") {
    std::string path;
    std::cin >> path;
    std::ifstream in(path);
    if (in.is_open()) {
        int counter = 0;
        std::string x;
        while (in >> x && counter < 40) {
            if (x != "0" && x != "1") {
                std::cout << "Input Error";
                return -1;
            }
            Flowers[counter] = std::stoi(x);
            counter++;
        }
    }
}

```

```

        for (int i = 0; i < arrSize; i++) {
            Flowers[i] = rand() % 2;
            if (Flowers[i] == 0) {
                FadingFlowers.push_back(i);
            }
        }
    } else {
        std::cout << "Input Error";
        return -1;
    }
    in.close();
}

int num[2];
for (int i = 0; i < 2; i++) {
    num[i] = i;
}
//идентификаторы для дочерних потоков
pthread_t GardenerThread1;
pthread_t GardenerThread2;
//создание дочерних потоков
pthread_create(&GardenerThread1, NULL, Watering, (void
*) (num + 0));
pthread_create(&GardenerThread2, NULL, Watering, (void
*) (num + 1));
//Вызов
Fading(NULL);
pthread_join(GardenerThread1, NULL);
pthread_join(GardenerThread2, NULL);

pthread_mutex_destroy(&mutex);
}

```

Сравнительные результаты:

```
user@user-VirtualBox:~$ g++ main.cpp -o main
user@user-VirtualBox:~$ ./main
Enter "random", "direct" or "file": random
time = 2275 | Flowers: 0000001000011100001100000100111001001111
time = 2295 | Gardener #1 watered flower #2
time = 2304 | Gardener #2 watered flower #35
time = 2356 | Gardener #1 watered flower #1
time = 2380 | Gardener #2 watered flower #9
time = 2434 | Gardener #2 watered flower #6
time = 2454 | Gardener #1 watered flower #24
time = 2584 | Gardener #1 watered flower #16
time = 2626 | Gardener #2 watered flower #17
time = 2705 | Flowers: 1000011010011101101100010100111001100110
time = 2723 | Gardener #2 watered flower #23
time = 2729 | Gardener #1 watered flower #11
time = 2782 | Gardener #2 watered flower #18
time = 2801 | Gardener #1 watered flower #5
```

```
user@user-VirtualBox:~$ ./main random
time = 3324 | Flowers: 1001000100110011110100100001100001000000
time = 3367 | Gardener #2 watered flower #39
time = 3399 | Gardener #1 watered flower #3
time = 3469 | Gardener #2 watered flower #13
time = 3538 | Gardener #1 watered flower #5
time = 3602 | Gardener #2 watered flower #27
time = 3661 | Gardener #1 watered flower #22
```

```
user@user-VirtualBox:~$ ./main direct 0 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1
time = 2375 | Flowers: 0100101100001111110000000000000000000000
time = 2391 | Gardener #1 watered flower #9
time = 2402 | Gardener #2 watered flower #38
time = 2460 | Gardener #1 watered flower #28
time = 2482 | Gardener #2 watered flower #3
time = 2538 | Gardener #2 watered flower #23
time = 2561 | Gardener #1 watered flower #40
time = 2616 | Gardener #1 watered flower #36
time = 2639 | Gardener #2 watered flower #25
```



```
user@user-VirtualBox:~$ ./main
Enter "random", "direct" or "file": direct
Enter Flower[0], 0 - fading, 1 - beautiful: 1
Enter Flower[1], 0 - fading, 1 - beautiful: 0
Enter Flower[2], 0 - fading, 1 - beautiful: 1
Enter Flower[3], 0 - fading, 1 - beautiful: 0
Enter Flower[4], 0 - fading, 1 - beautiful: 1
Enter Flower[5], 0 - fading, 1 - beautiful: 0
Enter Flower[6], 0 - fading, 1 - beautiful: 1
Enter Flower[7], 0 - fading, 1 - beautiful: 0
Enter Flower[8], 0 - fading, 1 - beautiful: 0
Enter Flower[9], 0 - fading, 1 - beautiful: 0
Enter Flower[10], 0 - fading, 1 - beautiful: 0
Enter Flower[11], 0 - fading, 1 - beautiful: 1
Enter Flower[12], 0 - fading, 1 - beautiful: 1
Enter Flower[13], 0 - fading, 1 - beautiful: 1
Enter Flower[14], 0 - fading, 1 - beautiful: 1

user@user-VirtualBox:~$ ./main
Enter "random", "direct" or "file": file
.\inputfile.txt
Input Erroruser@user-VirtualBox:~$ ./main
Enter "random", "direct" or "file": file
/home/user/inputfile.txt
time = 2721 | Flowers: 0001010000000101001001011010100100011011
time = 2742 | Gardener #2 watered flower #11
time = 2750 | Gardener #1 watered flower #9
time = 2809 | Gardener #2 watered flower #7
time = 2832 | Gardener #1 watered flower #5
time = 2884 | Gardener #2 watered flower #30
time = 2902 | Gardener #1 watered flower #28
```

```
user@user-VirtualBox:~$ ./main file /home/user/newinputfile.txt
time = 2859 | Flowers: 1011111111100000011000000000000000000000
time = 2882 | Gardener #2 watered flower #33
time = 2893 | Gardener #1 watered flower #26
time = 2951 | Gardener #2 watered flower #35
time = 2974 | Gardener #1 watered flower #39
time = 3049 | Gardener #2 watered flower #20
time = 3071 | Gardener #1 watered flower #22
time = 3126 | Gardener #2 watered flower #27
time = 3148 | Gardener #1 watered flower #17
```

