НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии
Дисциплина: «Архитектура вычислительных систем»

Задача о клумбе.

Пояснительная записка

Выполнил:
Попов Андрей Эдуардович,
студент гр. БПИ197.

Москва
2020

1. Текст задания

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. Применяемые расчетные методы

2.1. Теория решения задания

Для решения поставленной задачи использовался следующий метод:

Два потока для изменения состояния цветов:
Поток “Садовник” – поливает цветы.
Поток “Бог Засухи” – высушивает цветы (меняет состояние цветка на «завял»).

Каждый поток-садовник ищет ещё не политый цветок и поливает его. После ищет следующий цветок с временной задержкой от 1 до 3 секунд. Использовались мутексы и защита записи, чтобы садовники не полили случайно один и тот же цветок.

Поток “Бог Засухи” ищет случайным образом ещё не политый цветок и меняет его состояние на «завял». После ищет следующий цветок с задержкой от 0.5 до 3 секунд. Также используются мутексы и защита записи.

Для того чтобы потоки ожидали, пока не появится хотя бы один увядший или хотя бы один политый цветок, используются семафоры.

2.2. Организация входных данных

Программа не запрашивает входные данные.

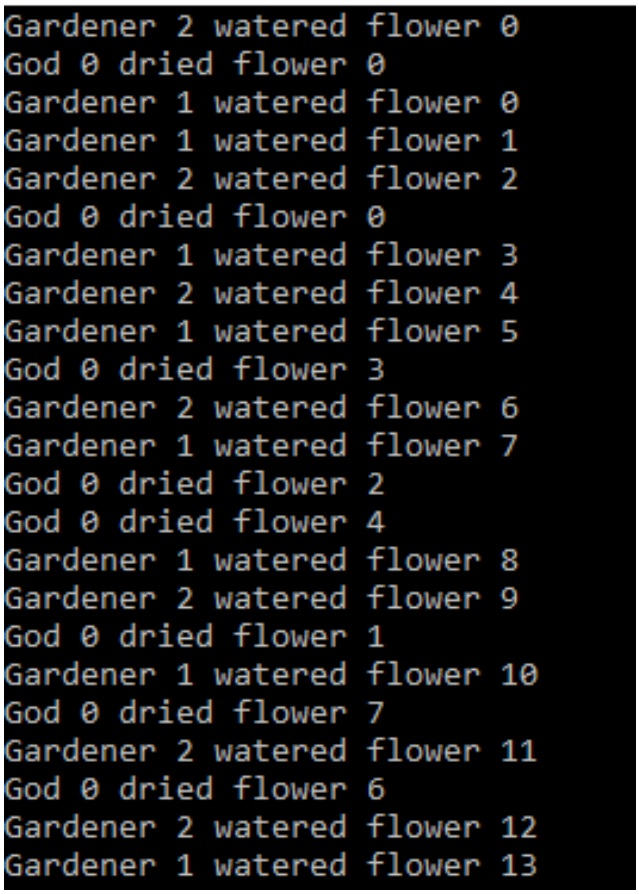
2.3 Организация выходных данных

Программа бесконечно сообщает пользователю о происходящем с клумбой. Т.е. сообщает, что Бог Засухи высушил какой-либо цветок или что какой-то из садовников полил какой-то цветок.

3. Тестирование программы

3.1. Корректные значения

Рисунок 1. Пример вывода информации о клумбе



3.2. Некорректные значения

В связи с отсутствием входных данных, нельзя ввести их некоректно

ПРИЛОЖЕНИЕ 1

Список литературы

1. Заголовок. [Электронный ресурс] // URL: http://www.softcraft.ru/edu/comparch/lect/07-parthread/multitreading.pdf

ПРИЛОЖЕНИЕ 2

Код программы

```
#define HAVE_STRUCT_TIMESPEC
#include <iostream>
#include <string>
#include <vector>
#include <ctime>
#include <pthread.h>
#include <thread>
#include <semaphore.h>
#include <future>

//Попов Андрей
//БПИ197
//Вариант 16
//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..


const int flower_count = 40;
sem_t watared; //семафор, отображающий о том как цветы политы
sem_t dried; //семафор, отображающий как цветы высушены
pthread_mutex_t mutexD; //мьютекс для операции записи
pthread_mutex_t mutexF; //мьютекс для операции чтения
bool flower_bed[flower_count];

/// <summary>
/// Возвращает случайное целое число включая обе границы
/// </summary>
/// <param name="min_value">левая граница</param>
/// <param name="max_value">правая граница</param>
/// <returns>Возвращает случайное целое число включая обе границы</returns>
int next_int(int min_value, int max_value, int my_seed = 0) {
	return min_value + ((rand() + my_seed) % (max_value - min_value + 1));
}

/// <summary>
/// Бог засухи (управляет увяданием цветов)
/// </summary>
/// <param name="param"></param>
/// <returns></returns>
```

```
void* God_Of_Drought(void* param) {
	int god_num = (int)param;
	while (true) {
		pthread_mutex_lock(&mutexD); //защита операции записи
		sem_wait(&watared); // уменьшить число политых цветоков на один
		int index = next_int(0, flower_count - 1);
		int count_search = 0;
		while (!flower_bed[index] && count_search++ < flower_count * 2) {
			index = next_int(0, flower_count - 1);
		}
		if (count_search < flower_count * 2) {
			flower_bed[index] = false; //критическая секция
			std::cout << "God " << god_num << " dried flower " << index << '\n';
		}
		sem_post(&dried); // число высушенных цветков увеличить на один
		pthread_mutex_unlock(&mutexD);
		std::this_thread::sleep_for(std::chrono::milliseconds(next_int(500, 3000)));
	}
	return nullptr;
}

/// <summary>
/// Садовник, противостоит богу засухи
/// </summary>
/// <param name="param"></param>
/// <returns></returns>
void* Gardener(void* param) {
	int gardener_num = (int)param;
	int flower_index = 0;
	while (true) {
		pthread_mutex_lock(&mutexD); //защита записи
		sem_wait(&dried); //количество высушенных ячеек уменьшить на единицу
		int count_search = 0;
		while (flower_bed[flower_index] && count_search++ < flower_count * 2) {
			flower_index = (flower_index + 1) % flower_count;
		}
		if (count_search < flower_count * 2) {
			flower_bed[flower_index] = true;
			std::cout << "Gardener " << gardener_num << " watered flower " <<
flower_index << '\n';
		}
		flower_index = (flower_index + 1) % flower_count; //критическая секция
		sem_post(&watared); //количество политых цветков увеличить увеличилось на 1
		pthread_mutex_unlock(&mutexD);
		std::this_thread::sleep_for(std::chrono::milliseconds(next_int(1000, 3000)));
	}
```

```
	return nullptr;
}

int main() {
	srand(time(NULL));
	//инициализация мьютексов и семафоров
	pthread_mutex_init(&mutexD, nullptr);
	pthread_mutex_init(&mutexF, nullptr);
	sem_init(&watared, 0, flower_count);
	sem_init(&dried, 0, flower_count);

	pthread_t god;
	pthread_t gardener1;
	pthread_create(&god, nullptr, God_Of_Drought, (void*)0); //Запуск бога засухи
	pthread_create(&gardener1, nullptr, Gardener, (void*)1);//запуск садовников
	Gardener((void*)2); //Один из садовников в основном потоке
	return 0;
}
```