

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет компьютерных наук, Департамент программной инженерии
Курс: «Архитектура вычислительных систем»

Микропроект 2

Многопоточное программирование. Взаимодействие потоков

Коновалов Егор Андреевич,
Студент БПИ195

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. Модель вычислений

Для реализации многопоточности используется C++ thread support library.

Для решения задачи программа использует $1 + n$ потоков, где n – количество садовников, следящих за клумбой (по умолчанию 2).

В главном потоке с заданным интервалом воспроизводится увядание случайного цветка следующим образом:

1. Из заданного числа цветков выбирается случайный
2. Выбранный цветок, представленный целым числом, добавляется в структуру данных очередь

Каждый поток садовника в необходимый момент поливает первый в очереди цветок, удаляя значение, представляющее его, из очереди.

Синхронизация потоков и управление доступом к данным осуществляется за счет примитива синхронизации – условная переменная, с помощью мьютексов.

При запуске потоков-садовников они становятся в очередь в состояние ожидания сигнала (метод `wait()`), не блокируя доступ к данным. Когда на главном потоке имитируется увядание цветка, ожидающим потокам подаётся сигнал (метод `notify_one()`), выводящий из ожидания первый поток в очереди. Первый поток имитирует полив цветка, после которого засыпает, освобождая доступ к данным. По окончании сна, поток снова встанет в очередь. При следующем увядании, будет снова вызван первый в очереди поток.

При каждом выходе из ожидания поток проверяет, что очередь цветков не пуста, для исключения ложного сигнала.

Поток-садовник выполняется до тех пор, пока не достигнуто заданное максимальное количество увяданий за время работы программы, либо пока очередь увядших цветков не пуста.

При достижении максимального числа увяданий за время выполнения программы, работа на главном потоке останавливается и все потоки-

садовники уведомляются (метод `notify_all()`) для гарантированного выхода их из состояния ожидания.

3. Параметры модели

Ключевые параметры модели могут быть заданы через аргументы программы. Их порядок, описание и значения по умолчанию приведены в таблице:

Название	Описание	Значение по умолчанию
flowerCount	Количество цветков в клумбе	40
gardenerCount	Количество садовников	2
gardenerSleepMs	Длительность сна садовника в миллисекундах	500
flowerWitherIntervalMs	Количество миллисекунд до следующего увядания	1000
maxFlowerWithers	Максимальное количество увяданий за время выполнения программы	10

4. Используемые источники

1. Документация Thread support library
<https://en.cppreference.com/w/cpp/thread>
2. Условная переменная
https://ru.wikipedia.org/wiki/Условная_переменная
3. Материал курса «Архитектура вычислительных систем»
<http://www.softcraft.ru/edu/comparch/>
4. Уильямс Энтони. С++. Практика многопоточного программирования. — СПб.: Питер, 2020. — 640 с.

5. Тестирование программы

1. Тестирование со значениями по умолчанию (след. страница):

```

C:\Users\user\Desktop\fasm\project02>ppmp.exe
0.0001398s: Flower 12 withered
0.0067468s: Gardener 0 watered flower 12
0.0075655s: Gardener 0 will be sleeping for 500ms
1.0012s: Flower 34 withered
1.0034s: Gardener 1 watered flower 34
1.00513s: Gardener 1 will be sleeping for 500ms
2.00361s: Flower 16 withered
2.30824s: Gardener 0 watered flower 16
2.31054s: Gardener 0 will be sleeping for 500ms
3.30864s: Flower 24 withered
3.31159s: Gardener 1 watered flower 24
3.31391s: Gardener 1 will be sleeping for 500ms
4.31237s: Flower 6 withered
4.3152s: Gardener 0 watered flower 6
4.31733s: Gardener 0 will be sleeping for 500ms
5.3157s: Flower 28 withered
5.31834s: Gardener 1 watered flower 28
5.32085s: Gardener 1 will be sleeping for 500ms
6.31993s: Flower 4 withered
6.32289s: Gardener 0 watered flower 4
6.32538s: Gardener 0 will be sleeping for 500ms
7.32332s: Flower 16 withered
7.32644s: Gardener 1 watered flower 16
7.32899s: Gardener 1 will be sleeping for 500ms
8.32793s: Flower 25 withered
8.32937s: Gardener 0 watered flower 25
8.33055s: Gardener 0 will be sleeping for 500ms
9.32977s: Flower 22 withered
9.3326s: Gardener 1 watered flower 22
9.3347s: Gardener 1 will be sleeping for 500ms
9.83743s: Gardener 1 finished his work
10.3333s: Gardener 0 finished his work

```

2. Тестирование с переданными аргументами:

```

C:\Users\user\Desktop\fasm\project02>ppmp.exe 20 5 200 500 8
0.0003324s: Flower 12 withered
0.0021527s: Gardener 3 watered flower 12
0.0034171s: Gardener 3 will be sleeping for 200ms
0.50174s: Flower 14 withered
0.503875s: Gardener 0 watered flower 14
0.505756s: Gardener 0 will be sleeping for 200ms
1.00513s: Flower 16 withered
1.00784s: Gardener 1 watered flower 16
1.01079s: Gardener 1 will be sleeping for 200ms
1.50859s: Flower 4 withered
1.51164s: Gardener 4 watered flower 4
1.51531s: Gardener 4 will be sleeping for 200ms
2.01232s: Flower 6 withered
2.01457s: Gardener 2 watered flower 6
2.01739s: Gardener 2 will be sleeping for 200ms
2.51506s: Flower 8 withered
2.51751s: Gardener 3 watered flower 8
2.52087s: Gardener 3 will be sleeping for 200ms
3.01824s: Flower 4 withered
3.02068s: Gardener 0 watered flower 4
3.02424s: Gardener 0 will be sleeping for 200ms
3.52072s: Flower 16 withered
3.52296s: Gardener 1 watered flower 16
3.52565s: Gardener 1 will be sleeping for 200ms
3.7277s: Gardener 1 finished his work
4.02351s: Gardener 0 finished his work
4.02594s: Gardener 3 finished his work
4.02855s: Gardener 2 finished his work
4.03166s: Gardener 4 finished his work

```

3. Ввод неверных аргументов

```
C:\Users\user\Desktop\fas\project02>ppmp.exe 20 5 200 500 -1
Argument format: <flowerCount> <gardenerCount> <gardenerSleepMs>
<flowerWitherIntervalMs> <maxFlowerWithers>
Every argument must be greater than or equal 1
```