

# ООП

## Семестр 2. Лекция 6. Потоки

Кафедра ИВТ и ПМ  
ЗабГУ

2018

# План

Процессы и потоки

Потоки в стандартной библиотеке

Потоки в Qt

Потоки в C#

**Поток** выполнения (тред; thread — нить) — наименьшая единица обработки, исполнение которой может быть назначено ядром операционной системы.

# Outline

Процессы и потоки

Потоки в стандартной библиотеке

Потоки в Qt

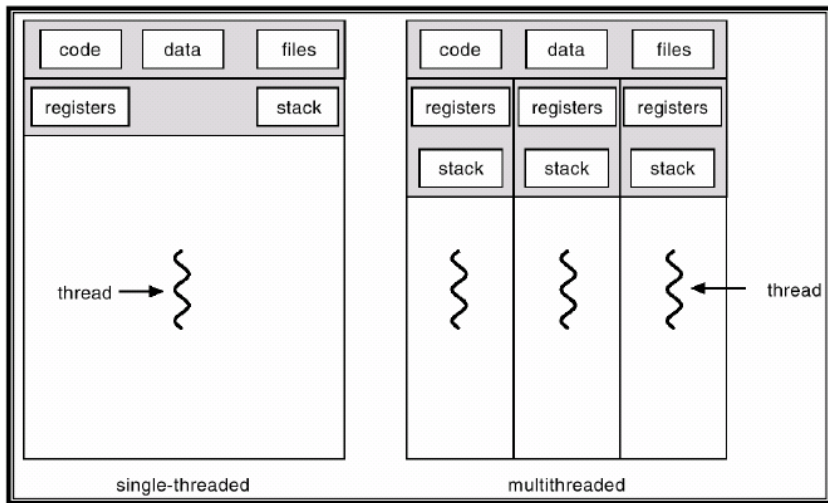
Потоки в C#

# Процесс

**Процесс** — программа, которая выполняется в текущий момент.

# Процессы и потоки

Один процесс может состоять из нескольких потоков.



# Многопоточность

**Многопоточность** — свойство платформы (операционной системы, виртуальной машины и т. д.) или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков, выполняющихся «параллельно», то есть без предписанного порядка во времени.

# Многопоточность

Многопоточность может быть полезна например, когда программа занята продолжительными вычислениями и одновременно должна отвечать на действия пользователя.

Поток программы будет занимается продолжительными вычислениями, а другой, основной - обработкой событий пользователя.



# Outline

Процессы и потоки

Потоки в стандартной библиотеке

Потоки в Qt

Потоки в C#

## std::thread

- ▶ `#include <thread>`

- ▶ `std::thread`

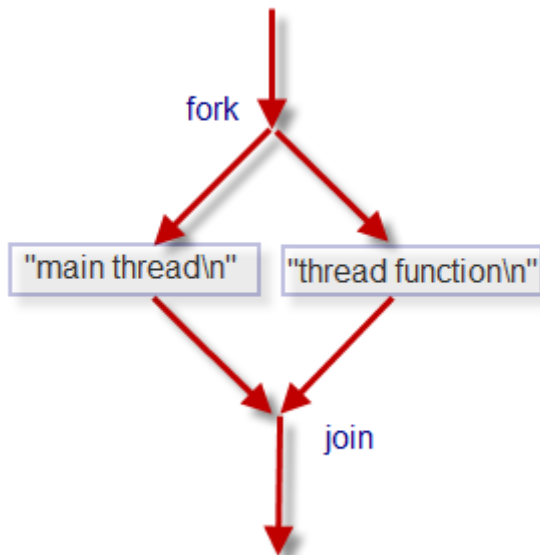
этот класс отвечает за работу с потоками

- ▶ `std::thread my_thread( <function>, [params] )`

новый поток стартует как только создан объект

- ▶ `my_thread.join()`

Если нужно дождаться завершения потока, то вызывается этот метод. Метод будет завершён как только завершится выполнение потока.



## std::thread

ошибка: undefined reference to `pthread\_create'

Для использования потоков нужно подключить к проекту соответствующую скомпилированную библиотеку.

- ▶ В Qt Creator добавить в файл проекта:

```
LIBS += -lpthread
```

или

```
LIBS += -pthread
```



## std::thread

```
#include <iostream>
#include <thread>
#include <math.h>
using namespace std;
void foo(unsigned long n, char c){
    float sum = 0;
    for (unsigned i = 0; i<n; i++){
        sum += sin(i);} }

int main(){
    unsigned long N = 300000000;
    // первый параметр - функция
    // остальные - параметры функции
    // потоки запускаются немедленно
    // два вызова функции foo будут выполнены параллельно
    std::thread th1(foo, N, '|');
    std::thread th2(foo, N, '.');
    // ожидание завершения первого, _затем_ второго потока
    th1.join();
    th2.join();
    return 0;
```

```
}
```

# Outline

Процессы и потоки

Потоки в стандартной библиотеке

Потоки в Qt

Потоки в C#

# QThread

В Qt принят объектно-ориентированный подход при работе с потоками.

- ▶ Вместо отдельной функции, которая должна выполняться в отдельном потоке используется метод.
- ▶ Выполнение отдельного метода потоку не назначается, вместо этого назначается выполнение методов конкретного *объекта*.

```
worker->moveToThread( thread );
```

# QThread

- ▶ Таким образом вместо создания функции требуется создать класс.
- ▶ Методы класса соединяются с помощью механизма сигналов и слотов с методами потока:  
запуск потока -> запуск метода  
завершения работы метода -> остановка вычислений.

```
connect(thread, &QThread::started,  
        worker, &Worker::process);
```

```
connect(worker, &Worker::finished,  
        thread, &QThread::quit);
```

worker - объект "работающий" в отдельном потоке.



# QThread

За работу с потоками в Qt отвечает класс QThread.

```
#include <QThread>
```

```
...
```

```
QThread *th = new QThread();  
Worker *worker = new Worker();
```

```
worker->moveToThread(th);
```

```
// Запуск потока должен запустить вычисления в классе  
connect(th, &QThread::started, worker, &Worker::process);
```

```
// после завершения работы, метод process класса Worker  
// должен вызывать сигнал finished  
// Который должен быть соединён с методом остановки потока.  
connect(worker, &Worker::finished, th, &QThread::quit);
```

см. пример example\_qthread [github.com/VetrovSV/OOP](https://github.com/VetrovSV/OOP)

В примере `example_qthread` можно пойти дальше и разделить *вычисления* на несколько потоков.

Каждый поток будет вычислять свою часть суммы. Например если использовать четыре потока, то каждый поток будет вычислять сумму для каждого 4-го значения  $i$ . Начальными же значениями  $i$  для каждого из потоков будут 0,1,2 и 3 соответственно.

# Outline

Процессы и потоки

Потоки в стандартной библиотеке

Потоки в Qt

Потоки в C#

## Потоки в C#. Пример

```
static void mythread1(){
    for (int i = 0; i < 10; i++){
        Console.WriteLine("Поток 1 выводит " + i);}
}

static void mythread2(){
    for (int i = 0; i < 10; i++){
        Console.WriteLine("Поток 2 выводит " + i);}
}

static void Main(string[] args){
    Thread thread1 = new Thread(mythread1);
    Thread thread2 = new Thread(mythread2);

    thread1.Start();
    thread2.Start();

    for (int i = 0; i < 10; i++){
        Console.WriteLine("Поток 3 выводит " + i);}
    Console.ReadLine();
}
```

## Потоки в С. Пример

```
Поток 3 выводит 0
Поток 3 выводит 1
Поток 3 выводит 2
Поток 3 выводит 3
Поток 3 выводит 4
Поток 3 выводит 5
Поток 3 выводит 6
Поток 1 выводит 0
Поток 1 выводит 1
Поток 1 выводит 2
Поток 1 выводит 3
Поток 1 выводит 4
Поток 1 выводит 5
Поток 1 выводит 6
Поток 1 выводит 7
Поток 1 выводит 8
Поток 1 выводит 9
Поток 2 выводит 0
Поток 2 выводит 1
Поток 2 выводит 2
Поток 2 выводит 3
Поток 2 выводит 4
Поток 2 выводит 5
Поток 2 выводит 6
Поток 2 выводит 7
Поток 2 выводит 8
Поток 3 выводит 7
Поток 3 выводит 8
Поток 3 выводит 9
Поток 2 выводит 9
```

## Ссылки и литература

1. [doc.qt.io/qt-5/qthread.html](http://doc.qt.io/qt-5/qthread.html) - документация QThread
2. [habr.com/post/150274](http://habr.com/post/150274) - Правильное использование QThread
3. [habr.com/post/182610](http://habr.com/post/182610) - Потоки, блокировки и условные переменные в C++11
4. [C++11/C++14 Thread Tutorials](#)
5. [vscode.ru/prog-lessons/potoki-v-si-sharp.html](http://vscode.ru/prog-lessons/potoki-v-si-sharp.html) - Потоки в C для начинающих: разбор, реализация, примеры

## Ссылки и литература

1. Г. Буч. Объектно-ориентированный анализ и проектирование с примерами приложений. 720 с. 2010 г. 700 страниц. Теория. Примеры на C++. Картинки! Вторая половина книги - примеры OOA и OOD с UML диаграммами.
2. MSDN - Microsoft Developer Network
3. Qt 5.X. Профессиональное программирование на C++. Макс Шлее. 2015 и более поздние издания г. 928 с. Книга периодически обновляется с выходом новых версий фреймворка Qt.
4. [www.stackoverflow.com](http://www.stackoverflow.com) - система вопросов и ответов
5. [draw.io](http://draw.io) — создание диаграмм.

# Материалы курса

Слайды, вопросы к экзамену, задания, примеры

[github.com/VetrovSV/OOP](https://github.com/VetrovSV/OOP)