

ООП

Семестр 2. Лекция 6. Потоки

Кафедра ИВТ и ПМ
ЗабГУ

2018

План

Процессы и потоки

Потоки в стандартной библиотеке

Потоки в Qt

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

Outline

Процессы и потоки

Потоки в стандартной библиотеке

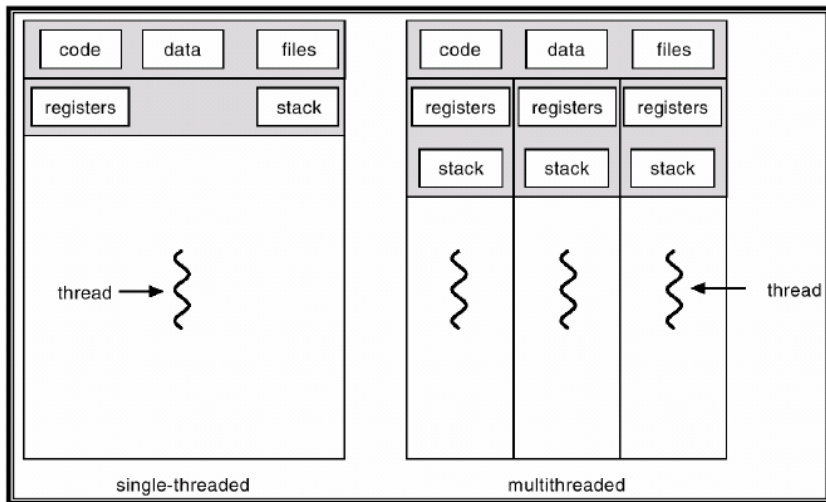
Потоки в Qt

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

std::thread

```
#include <thread>

void threadFunction()
{
    // do smth
}

int main()
{
    std::thread thr(threadFunction);
    thr.join();
    return 0;
}
```

Outline

Процессы и потоки

Потоки в стандартной библиотеке

Потоки в Qt

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

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.

Ссылки и литература

1. doc.qt.io/qt-5/qthread.html - документация QThread
2. habr.com/post/150274 - Правильное использование QThread
3. habr.com/post/182610 - Потоки, блокировки и условные переменные в C++11

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 - система вопросов и ответов
5. draw.io — создание диаграмм.

Материалы курса

Слайды, вопросы к экзамену, задания, примеры

github.com/VetrovSV/OOP