



Studio Mobile

Многопоточное программирование в iPhone

Андрей Вербин

andrey@studiomobile.ru

<http://studiomobile.ru>



Studio Mobile

МНОГОПОТОЧНОСТЬ

- Горячая тема, множество библиотек, проектов и идей
- Intel Threading Building Blocks
- Microsoft RX
- Функциональные языки
- Многопоточность сегодня решает две задачи:
- Распределения нагрузки между несколькими ядрами процессора
- Асинхронные вызовы



Studio Mobile

Многопоточность в iPhone

- iPhone - только одно ядро
- Бесполезно пытаться увеличить производительность программы с помощью нескольких потоков.
- Но можно скрыть задержки вызванные
 - Сетью
 - Сложными запросами в базу данных
 - Большими файлами
 - Любыми другими долгими операциями



Studio Mobile

Проблема



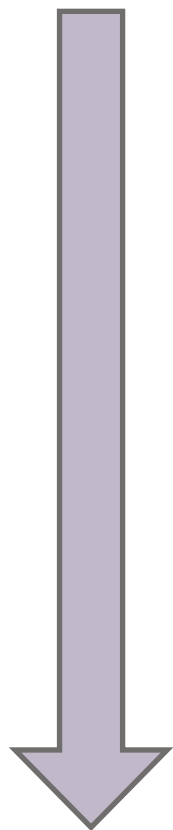


Studio Mobile

Инструментарий

Уровень абстракции

Простота использования



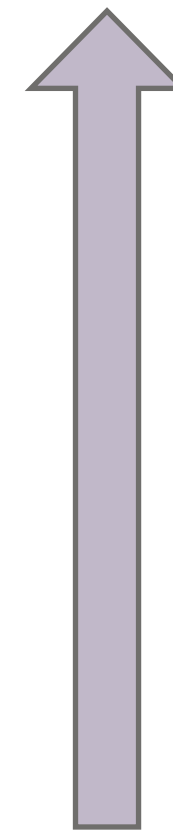
NSOperationQueue

Grand Central Dispatch

NSThread/pthread



kernel/mach threads





Studio Mobile

NSThread/pthread

- Комбинация NSThread и механизмов блокировки позволяет реализовать любые сценарии
- Сложно использовать
- Дедлоки - два потока бесконечно долго ждут друг друга
- Гейзенбаги (в честь принципа Гейзенберга) - ошибки, которые воспроизводятся иногда. Как правило чаще всего в руках пользователя :(
- Нужно изучить десяток различных примитивов синхронизации: блокировки, семафоры, мониторы, барьеры и тд.
- Инструмент для крайнего случая



Studio Mobile

NSThread/pthread

- performSelectorInBackground:withObject:

```
- (void)performSelector {  
    [self performSelectorInBackground:@selector(doSomeBackgroundWork):  
withObject:@"Very important data"];  
}
```

- performSelectorOnMainThread:withObject:waitUntilDone:

```
- (void)doSomeBackgroundWork:(NSString*)data {  
    NSLog(@"Background: %@", data);  
    [self performSelectorOnMainThread:@selector(handleResults):  
withObject:data waitUntilDone:YES];  
}  
  
- (void)handleResults:(NSString*)results {  
    NSLog(@"Main thread:%@", results);  
}
```



Studio Mobile

NSThread/pthread

Потоки не бесплатны!

Структуры ядра	1Kb (большая часть этой памяти не может быть сброшена в файл подкачки)
Стек	1Mb (это значение можно контролировать с помощью [NSThread setStackSize:])
Время создания	90 микросекунд

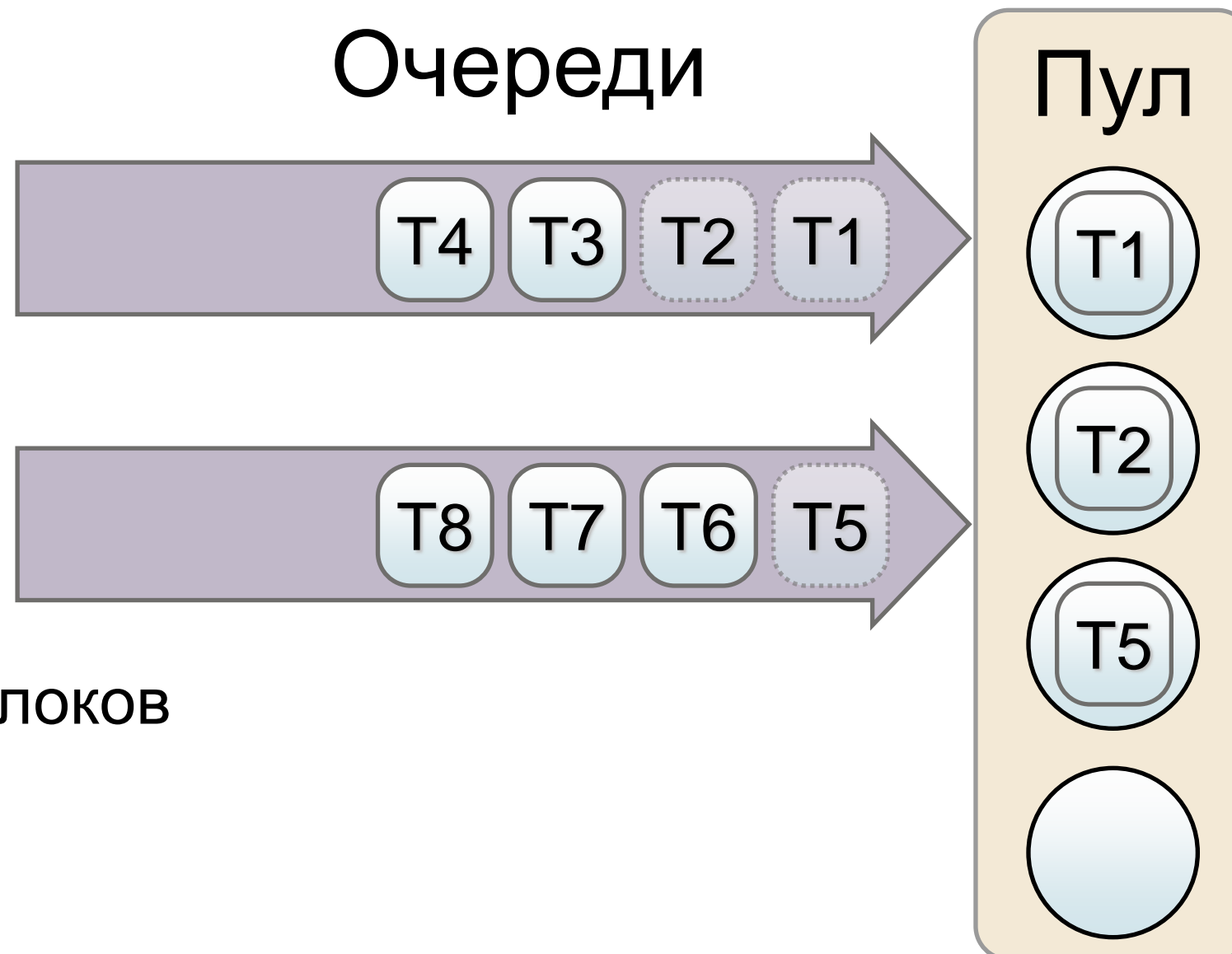
Их сложно использовать для выполнения большого количества небольших задач!



Studio Mobile

Grand Central Dispatch

- Пул потоков
- Очереди
- Асинхронные



- Синхронные
- Группы задач
- Задачи в виде блоков
 $\wedge\{\dots\}$



Studio Mobile

Grand Central Dispatch

Синхронизировать две задачи

```
//создать очередь
dispatch_queue_t queue = dispatch_queue_create("this.is.my.queue", NULL);
//и группу
dispatch_group_t group = dispatch_group_create();
//запустить две задачи
dispatch_group_async(group, queue, ^{
    NSLog(@"Some part of work done");
});
dispatch_group_async(group, queue, ^{
    NSLog(@"Another part of work done");
});
//дождаться завершения группы и выполнить код в главном потоке
dispatch_group_notify(group, dispatch_get_main_queue(), ^{
    NSLog(@"Finished");
});
```



Studio Mobile

Grand Central Dispatch

Близок к идеалу, но еще не совсем

- Низкоуровневый
- Сложно отменять задачи
- Сложно выделять задачи в отдельные модули/классы



Studio Mobile

NSOperationQueue

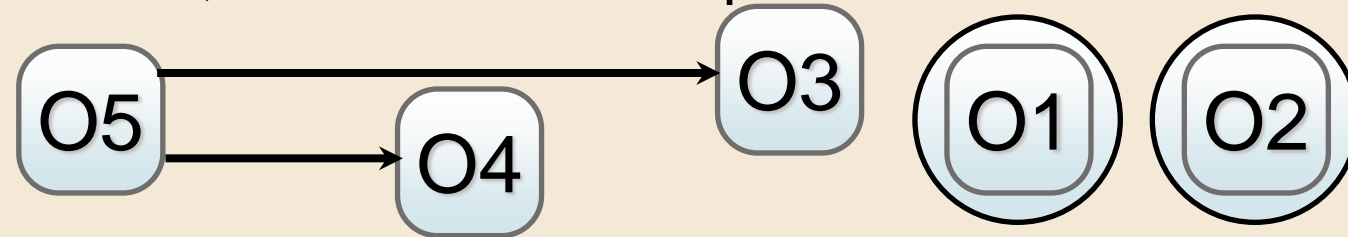
- Механизм основан на GCD
- Можно отменять операции
- Каждая операция может быть выделена в отдельный класс
- Между операциями можно устанавливать зависимости



Studio Mobile

NSOperationQueue

NSOperationQueue, maxConcurrentOperationCount = 2



```
NSOperationQueue *queue = [[NSOperationQueue alloc] init];
queue.maxConcurrentOperationCount = 2;
NSOperation *o1 = ..., *o2 = ..., *o3 = ..., *o4 = ..., *o5 = ...;
[o5 addDependency:o3];
[o5 addDependency:o4];
[queue addOperation:o1];
[queue addOperation:o2];
[queue addOperation:o3];
[queue addOperation:o4];
[queue addOperation:o5];
```



Studio Mobile

Типы операций

NSOperation	<pre>NSOperation *op = [RSSFetchOperation new]; [queue addOperation:op];</pre>
NSInvocationOperation	<pre>NSInvocationOperation *op = [[NSInvocationOperation alloc] initWithTarget:t selector:@selector(load) object:nil]; [queue addOperation:op];</pre>
NSBlockOperation	<pre>NSOperation *op = [NSBlockOperation blockOperationWithBlock:^(NSLog(@"Working in background!");)]; [queue addOperation:op];</pre>



Studio Mobile

Заключение

- Пример: RSS читалка новостей BBC.
https://github.com/studiomobile/rss_reader_demo
- Демонстрирует использование NSOperationQueue для загрузки контента из сети
- В отличие от примеров Apple, демонстрирует основательный подход к архитектуре приложения
- Большая тема, о которой мы не говорили - асинхронный ввод/вывод



Studio Mobile

Спасибо за
внимание!

Андрей Вербин

andrey@studiomobile.ru

<http://studiomobile.ru>