

iOS开发中多线程技术的研究和实践

黄浏展

(武汉职业技术学院计算机与软件工程学院, 湖北 武汉 430074)

摘要: iOS平台为适应多核心处理器的发展, 对多线程技术提供了很好的支持。在iOS应用开发中灵活运用多线程技术可以提高iPhone处理器的利用率, 最终提高手机运行效率, 给用户更好的体验。通过介绍多线程技术的相关知识, 阐述了多线程开发的优缺点, 列举了iOS支持的多种多线程技术。最后, 结合实例重点阐述了GCD多线程技术在iOS开发中的应用, 对理解和掌握iOS多线程开发有一定的帮助。

关键词: 多线程; iOS; 开发; GCD

中图分类号: TP311.1 **文献标识码:** A

Research and Practice of multi-thread Technology in IOS Development

HUANG Liuzhan

(School of Computer and Software Engineering, Wuhan Polytechnic, Wuhan 430074, China)

Abstract: The iOS platform provides a good support for multi-thread technology in order to adapt to the development of multi-core processors. Flexible use of multi-thread technology in iOS application development can improve the utilization efficiency of the iPhone processor, ultimately improving the efficiency of mobile phone operation, and give users a better experience. By introducing the related knowledge of multi-thread technology, this paper expounds the advantages and disadvantages of multi-thread development, and lists a variety of multi-thread technologies supported by iOS. Finally, the application of GCD multi-thread technology in iOS development is illustrated with an example, which is helpful to understand and master iOS multi-thread development.

Keywords: multi-thread; iOS; development; GCD

1 引言(Introduction)

随着用户对计算机计算能力需求的不断追求, 为了适应用户的更高要求, 多核心处理器已被广泛使用在各类电子设备上, 手机也不例外。多核心处理器使手机拥有同时执行多个任务的能力。但并非硬件上拥有高速多核心处理器, 手机就能发挥高性能, 还必须有软件相应支持, 多处理器的性能瓶颈在软件上^[1]。

多核环境下软件开发的核心理念是多线程开发。多线程技术不仅要求软件实现多线程, 在硬件上也必须采用多线程技

术。只有与多核CPU相适应的软件, 才能真正发挥多核的性能。iOS平台对多线程技术提供了很好的支持, 大多基于ios平台开发的app都使用了多线程技术, 下面将对ios开发中多线程技术的应用进行介绍。

2 多线程介绍(Introduction to multi-thread)

2.1 线程和多线程

要了解什么是线程, 首先要知道进程的概念。手机上每一个应用在运行前都会进驻内存, 进入内存并且运行中的应用程序就可称为一个进程。一个进程可以包含一个或多个线

程。可以认为线程是进程在调度过程中能独立执行某项任务的最小单位。

多线程是能使多个线程并发执行的技术。应用程序在手机上运行时,经常会同时处理多项任务。在手机应用开发中,为了实现多个任务并发执行,就会采用多线程技术。一个线程同一时间只能处理一个任务,为了能同时执行多项任务,程序员编程过程中就会开辟多个线程,以此来提高资源的利用率,这就是多线程编程。对于多核处理器,可以调用多个处理器核心同时运行多个线程,从而达到同步完成多项任务。

2.2 多线程的优缺点

多线程开发技术是从单线程和多进程技术发展而来,相对于单线程和多进程而言,多线程开发有其优势,主要表现在以下几点。首先,单线程的任务只能一个个按顺序执行,因此,如果某个任务运行时间太长,就会影响到手机处理器的使用效率。而多线程技术可以实现多个任务同时处理,在多核处理器时代,处理器多个核心同时运行多个线程,处理多个任务,能大大提高资源的利用率。其次,相对于进程而言,线程可以利用其所在进程的资源,系统不需要单独为线程分配系统资源,对线程的调度所付出的开销比进程要小得多。同时能充分发挥多核心处理器的优势,真正实现多任务并发。而且,当硬件处理器数量增加时,不需要做任何更改,程序运行速度会相应变快。

多线程技术运用得当能给应用开发带来极大的好处,但其也存在一些缺点。开启的线程过多,处理器调度线程所耗资源就越大,程序设计也会更复杂。同时,开启线程会占用一定的内存空间,如果开启的线程数量过多,必将占用较大的内存空间,影响程序的性能。

3 iOS应用开发中的多线程技术(Multi-thread technology in iOS application development)

3.1 iOS中的多线程运行机制

iOS平台对多线程技术提供了很好的支持,在iOS应用开发中灵活运用多线程技术可以对手机应用进行很好的优化。对于手机而言,大多数app都是c/s模式,都会涉及UI和业务逻辑的并行运行问题。通常,进程在创建时都会同时运行一个线程,这个线程被称为主线程。iOS进程如果开启了多个线程,系统会以时间片轮转的方式,让各个线程轮流使用处理器资源,因为时间片的时间非常短,用户感觉象是同时在运行。最新的苹果X采用的六核心处理器,如果同时运行多个线

程,iOS可以将多个线程运行在多个核心上,实现真正的并发执行,提高处理器的效率,最终达到提高应用效率的目的。

iOS开发中,处理与UI相关的操作都会运行在主线程中,如显示或刷新界面,处理UI事件(点击、滚动、拖拽等)^[2]。对于耗时的操作都会放在子线程,比如从网上获取数据,因为网速、数据大小或者网络连接状态等因素都会影响到所需时间,这种操作放在主线程必然会造成主线程阻塞,影响应用运行的流畅度,给用户不好的体验。耗时操作放在子线程,则可以提高应用运行效率。

3.2 iOS平台主要的线程技术介绍^[3]

iOS支持四种多线程技术Pthread、GCD、NSThread、NSOperation,各技术的特点如图1所示。

名称	简介	语言
PThread	跨系统平台(Windows/Linux/Unix等),可移植,具有一套通用的多线程API,线程生命周期由程序员管理,使用难度大	c语言
GCD	苹果公司为多核并行运行所设计的解决方案,线程生命周期自动管理	c语言
NSThread	面向对象操作,由线程由程序员管理,可直接操作线程对象	objective-c
NSOperation	面向对象,是对GCD的一种封装,但比GCD多了一些简单使用的功能	objective-c

图1 多线程技术

Fig.1 Multi-thread technology

其中前面两种以C语言为基础,通过C语言来操作,对于不熟悉C语言编程的开发者而言,使用起来较困难。Pthread技术因为使用难度大,目前基本已经过时不用^[3]。而后两种则基于objective-c,通过objective-c类的方式来执行多线程操作。这两种方式以objective-c类的方式对线程进行管理和操作,操作面向对象,因此简单易用。但NSThread技术需要开发者自行管理线程生命周期及线程同步。当多个线程同时访问同一份资源时会造成数据错乱,必须使用同步锁来解决这个问题,会增加CPU资源的损耗。NSOperation技术基于GCD,并且多了一些简单易用的功能,但是因为它是对GCD的第二次封装,性能上比GCD略低^[4]。NSOperation是基于objective-c语言的多线程技术,并且是在GCD基础上发展而来,只要有了objective-c语言基础,掌握了GCD技术的原理,就很容易上手NSOperation技术。因此,下面将主要介绍GCD技术及其在开发中的应用。

4 GCD多线程开发技术(GCD multi-thread development technology)

GCD是为多核的并行运算而设计的纯C语言开发的一种

技术，提供了很多的功能强大的函数，利用这些可以创建线程、调度任务、销毁线程，开发者只需告诉GCD要执行的任务，不需要编写线程管理代码。在进行多线程开发时可以传Block也可以传c函数指针^[5]。在进行多核编程时方便高效，它可以自动利用更多的处理器核心进行工作。

4.1 队列和任务

要了解GCD的工作原理，首先要知道GCD的两个核心概念—队列和任务。实现多线程要做的两件事就是创建队列和提交任务。

队列(Dispatch Queue)是用来存放任务的集合并负责管理任务。队列的机制就是将任务拆分成多个工作单元，将其加入到队列中，系统会负责管理队列，并将队列放到多个线程上执行，开发者无需管理线程。

开发者可以使用系统提供的函数来创建队列，队列通过一个线程池来处理队列管理的任务。GCD的队列是严格遵守先进先出(FIFO)的原则，添加到队列的工作单元就是按照添加的循序启动。

根据任务执行方式的不同，队列可以分为串行队列和并行队列。串行队列的线程池中只有一个线程，队列中的任务按顺序一个个执行，每次只能执行一个任务。并发队列的线程池中有多个线程，因此并发队列的任务可以按先进新出的顺序并发启动，同时执行多个任务。

任务就是要执行的操作或实现的功能，任务通常通过同步函数和异步函数两种方式来执行。他们之间的区别在于只在当前线程中执行任务，不会开启新线程。而且，任务执行过程中，会阻塞当前线程。只有当前线程代码块任务执行结束，线程才会继续执行。而异步是在开辟新线程来执行任务，因此对当前线程不会有影响。

4.2 队列的创建

iOS创建队列大致可以分成三种类型：获取全局并发队列、创建串行和并行队列、获取主队列^[6]。开发者可以通过GCD提供的函数来创建和访问队列。下面将分别介绍这三种情况。

获取全局并发队列可以通过函数dispatch_get_global_queue(long identifier, unsigned long flags)来实现，该函数会返回一个dispatch_queue_t类型的对象，开发者只需通过该函数就可以获取系统给应用提供的多个并发的队列，这些队列在当前应用内全局共享。函数的第一个参数用于指定队列的优先级，查看官方文档可以得知优先级参数值。第二个参

数是供以后使用的，通常传入0即可

创建串行和并行队列用函数dispatch_queue_create(const char*_Nullable label,dispatch_queue_attr_t*_Nullable attr)来实现，同样，该函数也会返回一个dispatch_queue_t类型的对象。如果应用程序的任务需要按特定顺序执行，则创建串行队列。如果需要并发执行多个任务，可以通过获取全局并发队列。

创建串行或并发队列的函数中包含两个参数，第一个参数可以为队列设置一个字符串，也可以为NULL。第二个参数用来指定当前函数是创建串行队列还是并行队列，默认值为NULL，表示串行。

主队列是GCD中一个特殊的串行队列，开发者可以通过函数dispatch_get_main_queue(void)来获取主队列。应用中只要是提交给主队列的任务都会放在主线程中执行。

4.3 提交任务

创建完队列以后，需要将任务提交给队列。提交任务的方式分同步和异步两种。

开发者可以用两个函数dispatch_sync(dispatch_queue_t queue,DISPATCH_NOESCAPE dispatch_block_t block)和dispatch_sync_f(dispatch_queue_t queue,void*_Nullable context,dispatch_function_t work)来实现同步执行任务。这两个函数的区别在于前者是将要执行的任务以代码块的形式提交给指定的队列，后者是将要执行的任务以函数的方式提交给指定的队列。

在开发过程中，可以任意选择上述两个函数来实现任务同步。同步过程由GCD来处理，开发者只需将要执行的任务写入代码块或者要调用的函数中。与同步方式一样，异步执行任务也对应由两个函数来实现。分别是：dispatch_async(dispatch_queue_t queue,dispatch_block_t block),dispatch_async_f(dispatch_queue_t queue,void*_Nullable context,dispatch_function_t work)。两个函数的参数涵义和同步函数完全一致。前者是将代码块以异步的方式提交给指定的队列，后者是将函数以异步的方式提交给指定的队列。

4.4 队列和任务的组合

在开发过程中，可以根据实际情况，将不同的队列和不同的任务提交方式进行组合，包括串行同步、串行异步、并行同步、并行异步。采用同步还是异步决定是否需要开启新的线程，而任务的执行方式则取决于采用串行还是并行。不

同的组合方式将会产生不同的执行结果。

需要注意的是使用同步函数在当前串行队列中添加任务，会导致当前线程阻塞。而应用程序的主线程必须采取异步执行方式，这样才可以及时响应用户事件。

5 GCD多线程开发实践(GCD multi-thread development practice)

ios编程的工作原理示意图如图2所示，从图可以看出，手机端应用需要通过网络向服务器发送请求获取所需数据，这些数据可能包括图像、视频、文字等。获取到数据后再在UI上显示出来。这个请求过程所耗时间取决于网络服务和数据大小，在多线程编程中，网络数据请求通常会作为耗时操作而开启子线程。而UI显示操作，iOS要求必须在主线程^[7]。

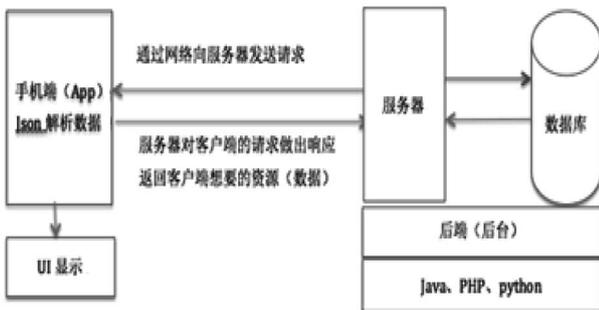


图2 app工作原理图

Fig.2 Working principle diagram

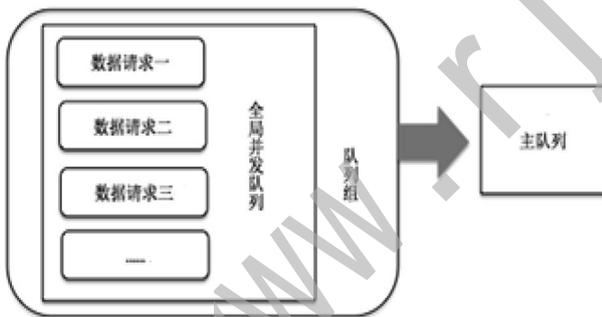


图3 多线程并发示意图

Fig.3 Multi thread concurrent schematic diagram

如果同一个界面需要通过网络请求从服务器获取多组数据，则可以通过GCD队列组来实现。首先使用异步执行多个数据请求操作，并在异步操作都执行完毕后，回到主线程执行操作。可以将多个block组成一个队列组，用于监听所有数据请求是否完毕，如果所有数据请求已完成则转到主线程执行UI操作(图3)。代码如下。

```
-(void)dispatch_group_requestData
{
    //调度组
    dispatch_group_t group=dispatch_group_create();
```

```
dispatch_group_async(group,dispatch_get_global_queue(0,0),^
    // 异步请求数据1代码 );
    dispatch_group_async(group,dispatch_get_global_queue(0,0),^
    // 异步请求数据2代码);
    dispatch_group_async(group,dispatch_get_global_queue(0,0),^
    // 异步请求数据3代码);
    dispatch_group_notify(group,dispatch_get_main_queue(),^
    // 数据请求完成，设置UI );};
```

6 结论(Conclusion)

随着手机性能不断增强，在ios应用开发过程中，必须注重对软件性能的优化，以便充分利用硬件资源，做到App运行整体性能的提升，给用户更好的体验。多线程开发技术提高了CPU的利用效率，但是线程的切换增加了系统开销，同时也面临着线程不安全、线程死锁的缺点[8]。因此，在ios开发中可以根据实际情况选择合适的多线程技术，但能单线程能处理的问题尽量不要使用多线程。

参考文献(References)

- [1] 眭俊华,刘慧娜,王建鑫,等.多核多线程技术综述[J].计算机应用,2013(33)(S1):239-242;261.
- [2] 孔翔鸣.Swift多线程编程[J].电脑知识与技术,2017(02):24-25.
- [3] 传智播客高教产品研发部.iOS开发项目化经典教程[M].北京:人民邮电出版社出版社,2016:1-5.
- [4] 陈伟,卜庆凯.iOS系统中多线程技术的研究[J].电脑知识与技术,2017(8):78-80.
- [5] 潘小龙.iOS系统中不同多线程技术的研究和比较[J].中国新通信,2014,16(24):21-22.
- [6] 李岚.iOS并发程序设计中几种方法的特点及使用技巧研究[J].电脑知识与技术,2016(9):83-84.
- [7] 孙翠改.Android环境下主UI线程与子线程通信机制研究[J].数字技术与应用,2016(9):66-67.
- [8] 崔政,段利国.基于Java synchronized同步锁实现线程交互[J].软件工程,2018(2):1-3.

作者简介:

黄浏展(1975-),男,硕士,副教授.研究领域:移动应用开发.