

个人电脑 计算机 计算机技术 多线程

关注者
1322

相关 Live 推荐

多线程有什么用？

最近看了多线程讲解很模糊

1 条评论 分享 邀请回答 举报 ...

关注问题

32 个回答

默认排序



pansz

Android 开发、Linux 话题的优秀回答者

1381 人赞同了该回答

这么解释问题吧：

- 1. 单进程单线程：一个人在一个桌子上吃菜。
- 2. 单进程多线程：多个人在同一个桌子上一起吃菜。
- 3. 多进程单线程：多个人每个人在自己的桌子上吃菜。

多线程的问题是多人同时吃一道菜的时候容易发生争抢，例如两个人同时夹一个菜，一个人刚伸出筷子，结果伸到的时候已经被夹走菜了。。。此时就必须等一个人夹一口之后，在还给另外一个

1. 对于 Windows 系统来说，【开桌子】的开销很大，因此 Windows 鼓励大家在一个桌子上吃菜。因此 Windows 多线程学习重点是要大量面对资源争抢与同步方面的问题。

2. 对于 Linux 系统来说，【开桌子】的开销很小，因此 Linux 鼓励大家尽量每个人都开自己的桌子吃菜。这带来新的问题是：坐在两张不同的桌子上，说话不方便。因此，Linux 下的学习重点大家要学习进程间通讯的方法。

--

补充：有人对这个开桌子的开销很有兴趣。我把这个问题推广说开一下。

开桌子的意思是指创建进程。开销这里主要指的是时间开销。

可以做个实验：创建一个进程，在进程中往内存写若干数据，然后读出该数据，然后退出。此过程重复 1000 次，相当于创建/销毁进程 1000 次。在我机器上的测试结果是：

UbuntuLinux：耗时 0.8 秒

Windows7：耗时 79.8 秒

两者开销大约相差一百倍。

这意味着，在 Windows 中，进程创建的开销不容忽视。换句话说就是，Windows 编程中不建议你创建进程，如果你的程序架构需要大量创建进程，那么最好是切换到 Linux 系统。

大量创建进程的典型例子有两个，一个是 gnu autotools 工具链，用于编译很多开源代码的，他们在 Windows 下编译速度会很慢，因此软件开发人员最好是避免使用 Windows。另一个是服务器，某些服务器框架依靠大量创建进程来干活，甚至是对每个用户请求就创建一个进程，这些服务器在 Windows 下运行的效率就会很差。这"可能"也是放眼全世界范围，Linux 服务器远远多于 Windows 服务器的原因。

--

再次补充：如果你是写服务器端应用的，其实在现在的网络服务模型下，开桌子的开销是可以忽略不计的，因为现在一般流行的是按照 CPU 核心数量开进程或者线程，开完之后在数量上一直保持，进程与线程内部使用协程或者异步通信来处理多个并发连接，因而开进程与开线程的开销可以忽略了。

另外一种新的开销被提上日程：核心切换开销。

现代的体系，一般 CPU 会有多个核心，而多个核心可以同时运行多个不同的线程或者进程。

当每个 CPU 核心运行一个进程的时候，由于每个进程的资源都独立，所以 CPU 核心之间切换的时候无需考虑上下文。

刘看山 · 知乎指南 · 知乎协
侵权举报 · 违法和不良信息
联系我们 © 2017 知乎

相关问题

当每个 CPU 核心运行一个线程的时候，由于每个线程需要共享资源，所以这些资源必须从 CPU 的一个核心被复制到另外一个核心，才能继续运算，这占用了额外的开销。换句话说，在 CPU 为多核的情况下，多线程在性能上不如多进程。

因而，当前面向多核的服务器端编程中，需要习惯多进程而非多线程。

编辑于 2014-07-24

▲ 1.4K ▼

💬 87 条评论

➦ 分享

★ 收藏

♥ 感谢

⋮

收起 ^

 **徐辰**
正在找能远程上班的工作

15 人赞同了该回答

因为你有多个CPU啊，一个线程只能用一个CPU，其它的放在那儿又不会生利息。
你说你只有一个CPU？那么多线程可以让它假装同时在干很多件事，也就是假装多个CPU，当然动作会慢一点，但很多时候你更慢对不。
哦，你说你就两个CPU为什么要创建500个线程？谁让你用Tomcat！

发布于 2016-03-29

▲ 15 ▼

💬 9 条评论

➦ 分享

★ 收藏

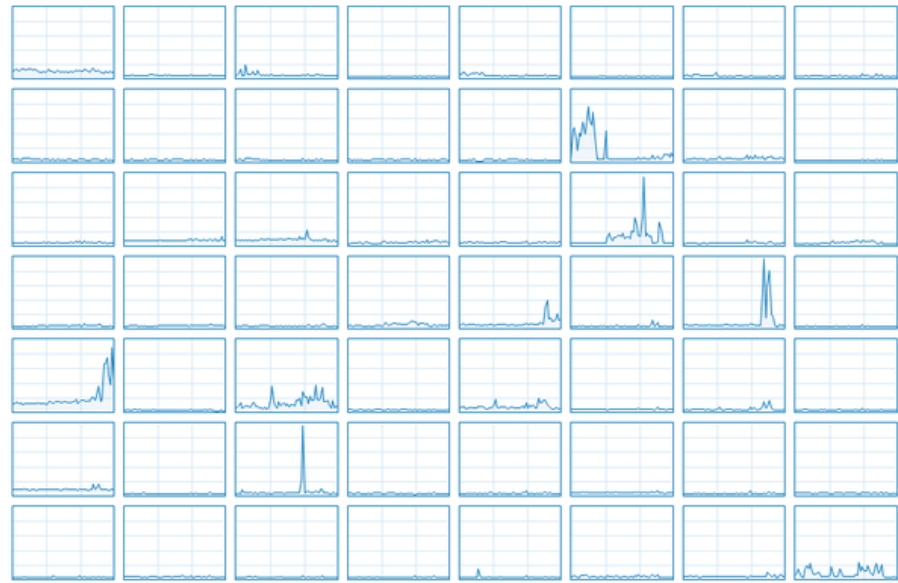
♥ 感谢

⋮

 **Belleve** 
编程 话题的优秀回答者

42 人赞同了该回答

使用多线程

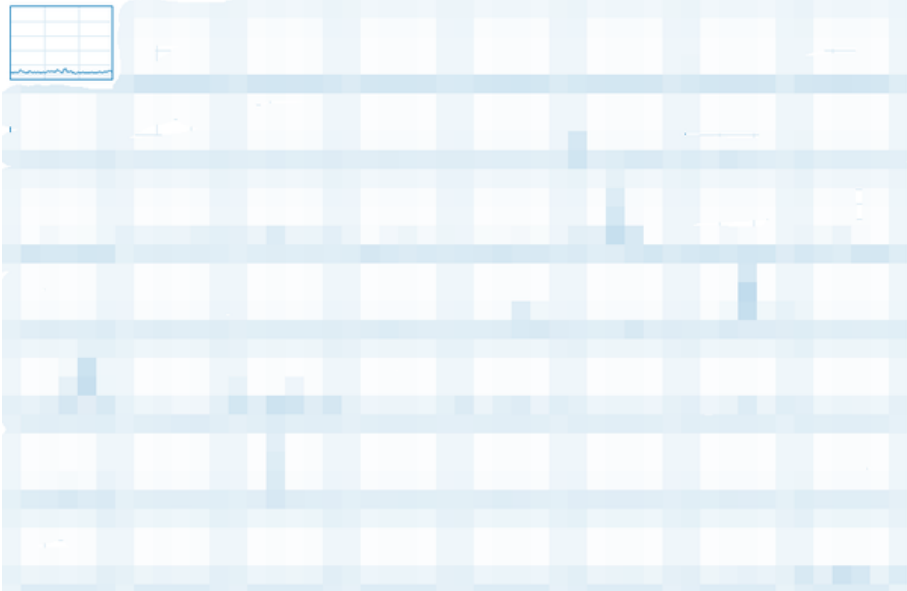


不用多线程

相关 Live 推荐

刘看山 · 知乎指南 · 知乎协
侵权举报 · 违法和不良信息
联系我们 © 2017 知乎

相关问题



发布于 2016-03-30

▲ 42 ▼ 13 条评论 分享 收藏 感谢 ... 收起 ^

 **yskin**
<--点赞的好帅！

110 人赞同了该回答

很多人答案说的是操作系统提供的多进程而不是单个程序内的多线程。

多线程使得程序内部可以分出多个线程来做多件事情，而不会造成程序界面卡死。比如迅雷等多线程下载工具就是典型的多线程。一个下载任务进来，迅雷把文件平分成10份，然后开10个线程分别下载。这时主界面是一个单独的线程，并不会因为下载文件而卡死。而且主线程可以控制下属线程，比如某个线程下载缓慢甚至停止，主线程可以把它强行关掉并重启另外一个线程。

另外就是一些程序的打印功能，比如记事本、Adobe Reader，打印的时候就只能打印，无法在主界面进行操作，而Word就有“后台打印”的功能，点了打印命令之后，还可以回到主界面进行修改、保存等操作。

另外多线程除了并行完成一些任务以外，还有生产者-消费者模式。比如Windows命令行下在某个硬盘根目录执行一个“dir/s | more”命令，前一条显示硬盘里的所有文件，要执行很久才能执行得完，后面那条命令会把前面命令的输出分屏显示出来。但是执行整条命令时，会立刻有显示，也就是说，前面一条命令输出满一页内容到缓冲区，more命令就把缓冲区封死了，等用户敲了一个键显示下一屏的时候，more命令把缓冲区的内容取出并清空，前面的命令才能输出下一屏到缓冲区。这样的多线程使得整条命令不用等待前面的命令全部执行完才能执行下一条命令。

多线程和多进程的区别。平常指的多进程是操作系统下同时运行多个进程，比如Word和Excel同时打开，并且可以并行地同时执行一些操作。这种多进程和多线程没什么好比较的。可以比较的是同一个程序里的多线程和多进程。

多线程因为在同一个进程里所以可以共享内存和其他资源，比如迅雷里10个线程一齐下载一个文件，这个文件是由进程打开的，然后10个线程都可以往里写入东西。如果是10个进程就不行了，操作系统不允许一个文件由两个进程同时写入。另外，Chrome就是一个典型的多进程程序，里面每个标签页、扩展、插件都是单独的进程，各自独占资源，相互隔离，一个进程出错死掉只会影响一个页面或者插件，再也不会出现Flash插件出错崩溃导致整个浏览器崩溃的情况了。

编辑于 2011-11-04

▲ 110 ▼ 6 条评论 分享 收藏 感谢 ...

 **开源哥**
入门级 Lisper，Java 痛恨者，FSF 会员

48 人赞同了该回答

我本来有一个问题
用多线程了后，现在两个题问了有我
(逃

相关 Live 推荐

刘看山 · 知乎指南 · 知乎协
侵权举报 · 违法和不良信息
联系我们 © 2017 知乎

编程到底难在哪里？ · 4
线程崩溃是否会造成过
答
开发多线程的程序应
10 个回答
想成为计算机技术高
吗？ 58 个回答
你碰到过的最难调试
的？ 924 关注问题

 一分钟读懂食
相关问题
 新环境下，人
 我是否该读研
 国际列车上的
 如何自学计算

发布于 2016-03-30

▲ 48

▼

8 条评论

分享

★ 收藏

♥ 感谢

...



耿龙

17 人赞同了该回答

使用多线程一般有两个不同的目的：

一是把程序细分成几个功能相对独立的模块，防止其中一个功能模块阻塞导致整个程序假死（GUI 程序是典型）

另一个就是提高运行效率，比如多个核同时跑，或者单核里面，某个线程进行IO操作时，另一个线程可以同时执行。

发布于 2015-03-06

▲ 17

▼

1 条评论

分享

★ 收藏

♥ 感谢

...



韩葆-逸松

七年软件质量与安全保障经验，Synopsys SIG客户经理(+Defensics,ProteCode)

15 人赞同了该回答

-楼上的同学已经把这个概念说的非常简单明了。我结合对几百个团队代码静态分析的经验说一说可能存在的问题。

同样以吃饭举例子，假设很多人需要吃饭吃饭，桌子不够，那么桌子就是紧缺资源。单线程就是整个餐厅只有一个单人桌，这个人吃完了，下一个人轮上。但大餐馆用的可能是八仙桌(好吧我比较喜欢这种古代的方正桌子)，同时能容纳八个人吃饭，这就是多线程:从一次一个变成了一次多个或者多次多个。如何安排吃饭顺序对程序员来说是个巨大的挑战，主要问题在于对资源-桌子的安排上，举例说明:

1.死锁-大家都在等着吃饭，但需要特定条件才能开吃:桌一等着桌二的碗，桌二等着桌三的筷子，桌三呢，等着桌一的勺。于是这三桌就都不能吃饭。

2.非原子变量更新-空闲桌子的数字是要加以保护的，类似于特定人员才能修改，假如没有保护，空桌子一会儿是四个，一会儿是五个，最终的数字往往是不正确的，服务员要被玩死-这就是多线程的加锁操作。

还有其他几种，就不多说了，多线程的调度和使用是一个坑，是否能用好是程序员水平高低的一个标志。

发布于 2016-03-30

▲ 15

▼

添加评论

分享

★ 收藏

♥ 感谢

...



郭靖

知行合一 行胜于言·特别能吃苦，特别能战斗

我的理解是一个程序（载入内存被cpu执行的了就是进程了）的多个部分同时执行的话，那么就会更快的完成，尤其是多个部分是需要协作的，这个部分出了结果，那个部分刚好执行到要用这个结果，那感觉不能更爽。之前的单核时代，cpu还要在不同部分跳来跳去的，现在的多核时代，一个cpu跑一段，爽得不能更爽。这是我学的时候写的一篇博客，一个很简单的例子，但是把用处说清楚了。

[c# multi thread programming ,c# 多线程编程](#)

如果有操作系统的基本概念，这个就是不会模糊了。把程序想象成一个长条的0110序列在内存中，cpu在上面跑。

编辑于 2017-06-11

▲ 0

▼

添加评论

分享

★ 收藏

♥ 感谢

...



Pluto Hades

学而不思则罔，思而不学则殆。

12 人赞同了该回答

我从计算机发展历史上简单介绍一下。

一开始的计算机，只有一个处理器，处理器也没有多核多线程的概念。干的活都是人一条条指令安排好，依次去干。这就是最初的单任务处理器。

后来慢慢的，自然会产生有时候需要有多件事情要干的情况。比如，有ABC三件事情，我们把他们

相关 Live 推荐

刘看山 · 知乎指南 · 知乎协
侵权举报 · 违法和不良信息
联系我们 © 2017 知乎

相关问题

都分成一小块一小块，干一会A再干一会B再干一会C。于是就诞生了最初的协作模式和抢占模式，分时处理和实时处理。多任务模式和多任务处理器产生了，压栈弹栈也应用了。但是最初的安排都是大家一起协调的，人为把代码按功能分割开来。因为那时候还没有完整操作系统概念呢。后来，大家觉得，卧槽，这样安排太费时费力，我们应该让计算机自己处理这个，我们得把精力集中在干我们的正事儿上。于是，设计了操作系统来管理资源，引入线程来作为多任务抽象。再后来，大家觉得操作系统是好的，于是处理器的设计也开始为了便于操作系统管理而增加了更多任务管理的功能。慢慢的更多任务调度的数学模型被建立和引入，逐渐形成现在的体系。其实计算机的历史虽然不长，但是很有趣的。读读历史，会使人了解为什么世界是现在这个样子而不是别的样子。

以下列举一些有助于理解上面这段的搜索关键词，每行一组：

单任务处理器
分时 实时 抢占 协作 多任务
操作系统发展史
8086汇编
80386汇编
实模式 保护模式
进程 线程 处理器调度
.....

发布于 2016-06-25

▲ 12 ▼

9 条评论

🔗 分享

★ 收藏

❤ 感谢

...



成晨
这不是很有趣嘛

3 人赞同了该回答

打个比方：我们寝室洗手台有2个水龙头，就是多线程。可以满足2个人刷牙。不然排队蛋疼。

发布于 2012-06-28

▲ 3 ▼

2 条评论

🔗 分享

★ 收藏

❤ 感谢

...



陆益凡
帅德补耀补药的。

增加资源利用率。

发布于 2016-03-30

▲ 0 ▼

添加评论

🔗 分享

★ 收藏

❤ 感谢

...



乔拉爵士
靠自己本事借的钱，凭什么让我还？

5 人赞同了该回答

充分利用现在的多核处理器，挖掘硬件上的潜力

发布于 2014-05-02

▲ 5 ▼

添加评论

🔗 分享

★ 收藏

❤ 感谢

...



jianlong liu
我雷电芽依要打死你 --no daemon

更模块化的编程模型和提升程序的数据吞吐量(好像是这么说的)

Ps. 还有多线程死机更爽

编辑于 2016-09-23

▲ 0 ▼

2 条评论

🔗 分享

★ 收藏

❤ 感谢

...



白起
围棋/游戏开发/象棋/web/创业/少儿教育

6 人赞同了该回答

相关 Live 推荐

刘看山 · 知乎指南 · 知乎协
侵权举报 · 违法和不良信息
联系我们 © 2017 知乎

相关问题

大家讲理论比较多，我来说点实际的使用场景，便于理解。

多线程常用于避免用户等待，提高响应速度，增强用户体验；这种场合下，对提高整体运算速度的帮助不大。

比如，客户端app中，需要下载一些数据的同时又能让用户继续操作，这时候再开一个线程来下载数据是常见的选择。这是iOS开发、android开发日常处理的事情。

发布于 2016-09-04

▲ 6 ▼

● 添加评论

➦ 分享

★ 收藏

♥ 感谢

...

 **dccmx**
搞技术的

13 人赞同了该回答

多线程是用来将计算并行化的。

举个最最最简单的例子，程序里面要干两件毫不相干的事情A和B，各费时10分钟，那单线程程序就会这样：

```
A ( ) ;  
B ( ) ;
```

等A完了再执行B，总共耗时20分钟。但如果将A和B用两个线程来做则如下：

```
线程A：A ( ) ;  
线程B：B ( ) ;  
等待两个线程结束。
```

如果两个线程被分配到不同的cpu上，则理论耗时共10分钟。

编辑于 2011-11-03

▲ 13 ▼

● 7 条评论

➦ 分享

★ 收藏

♥ 感谢

...

 **岡崎鏡**
天闻角川翻译 / Python码农 / 前四大狗

3 人赞同了该回答

说一点自己知道的，在**一个程序**中多线程有什么用。如有说的不对恳请指出。

一个常见但不算典型多线程应用的场景是，web程序处理请求。每个请求的处理可能需要走以下几步：

1. 收到请求，调用处理类/函数；
2. 分析请求，得知要获取的数据；
3. 连接数据库或者读取相关文件；
4. 计算得出结果；
5. 返回响应数据。

我们知道，无论你有多个线程，实际上最终决定并行计算量的是CPU的核心个数，只是CPU切换任务的速度对于人的感知来说相对快，所以在使用计算机时会觉得它能够同时做好多个事情。假如一个程序只有一个线程，它每次排队等候CPU使用时间都只有一个名额，自身的执行也要按顺序一步一步走下来。前一步未完成，下一步就不会执行。实际上很多时间我们并不需要等候上一步完成，就可以先行做下一步，以达到最大化利用计算机内的各种资源（不仅仅是CPU）来最快地完成任务的目的。

在上面的例子中，最能够反映这一点的就是连接数据库及读取文件的操作（尤其在读取多个文件时），即I/O操作。单次I/O操作相对来说是比较花时间的（即开销大），我们可以开一个新线程去连接、等候、获取数据然后返回。同时，让原有的线程继续执行不需要用到I/O结果的部分。等到I/O线程返回（读取完成的事件发生），再取得这部数据去进行相关的计算或者操作。

另一个例子，Python中有个名为paramiko的库，实现了ssh的功能。它通过ssh协议与远程机器的类建立连接的Transport类就用到了多线程：

```
class Transport (threading.Thread, ClosingContextManager):  
    """ An SSH Transport attaches to a stream (usually a socket),  
    negotiates an  
    encrypted session, authenticates, and then creates stream tunnels, called
```

相关 Live 推荐

刘看山 · 知乎指南 · 知乎协
侵权举报 · 违法和不良信息
联系我们 © 2017 知乎

相关问题

``channels <.Channel>`, across the session. Multiple channels can be multiplexed across the session. Instances of this class may be used as context managers. """`

...

当你需要同时向多台远程计算机传输文件时，可为每一台计算机建立一个Transport实例（即新开一个线程），然后同时进行传输。这样程序就不需要等到一台机器完成传输后再去下一台机器操作了。

编辑于 2016-06-25

▲ 3 ▼

● 1 条评论

➦ 分享

★ 收藏

♥ 感谢

...

收起 ^



lulu

用word写代码

就像厕所，四个马桶vs一个马桶

发布于 2016-03-30

▲ 0 ▼

● 1 条评论

➦ 分享

★ 收藏

♥ 感谢

...



岳先生

iOS

9 人赞同了该回答

为了讲清楚多线程编程，这儿先讲一个场景，

这个场景是这样子的，山上有座庙（**进程**），庙里住了很多小和尚和老和尚（**线程**），当然隔壁山上也有很多尼姑妹子。小和尚和老和尚们每天都需要下山挑水喝。这些个和尚有刚上山的（就是有点傻的），有比较聪明的，也有闹过矛盾的。场景，人物呢，大概就是这么个样子。

俗话说的好，有人的地方就有江湖。这些个小和尚，老和尚都不是很安分，为了每天的这个挑水工作闹了很多很多的矛盾，也犯过很多傻。

下面对挑水工作所有发生的情况做个详述：

- 1) 刚上山的小和尚挑水，就是比较傻的那个，和别人都不认识，每天挑水都自己一个人，一次挑一桶水，一天下来发现挑的水根本不够喝。（**串行**）
- 2) 和尚里面有个力气大的，和别人关系不好，每次挑水能挑三桶左右，一天下来，发现挑的水勉强够喝。（**并发**）
- 3) 和尚里面有几个关系好的，力气也比较大，每次挑水都一起去，每个人都能挑两桶左右，一天下来，发现挑的水不仅够喝，还有时间帮隔壁山尼姑妹子挑水。（**并行**）
- 4) 在水井里打水，有个专门的妹子负责打水，比较傻的那个小和尚每次都盯着打水的妹子，还随时随地看水桶装满没有（**同步**）；当然，有傻的，也有聪明的，这个聪明的和尚在妹子打水的时候，就乘机在一旁休息，水打满以后，妹子都过来告诉他。（**异步**）
- 5) 水井里有一天没水了，傻和尚看见水井里没水以后，在水井边上一直等着，等有水了再挑回去（**阻塞**）；聪明和尚呢，看见水井里没水了，一点都不犹豫的回去了。（**非阻塞**）
- 6) 又有一天，那闹过矛盾的两个和尚挑水，正好狭路相逢，谁也不让谁过去，一直僵持着。（**死锁**）
- 7) 寺庙里厨房空间有限，每次只能进去一定数量的和尚，当厨房满了以后，其他人只能在外面等着，直到里面的人出来。这个实现的方法是，厨房门口挂了很多锁，每个人进去以后都拿走一把锁，出来以后把锁挂上，这样外面的人就可以知道厨房里面人满没满。（**信号量**）

终于编完了。。。

通过上面生动形象的描述，这个线程里面的一些概念已经明白了。

下面解释下多线程。

还是上面的例子，在这个寺庙里面，如果每天都由那个新来的傻子和和尚挑水喝，肯定是不行的，有的喝没得喝咱且不说，你让他累死了咋整。何况寺庙里那么些个空着的水桶那么些个不干活的人闲着也是闲着。

相关 Live 推荐

刘看山 · 知乎指南 · 知乎创作中心 · 侵权举报 · 违法和不良信息 · 联系我们 © 2017 知乎

相关问题

所以多线程的意思就是多个线程一起去协同完成一个任务，通过充分去共享资源来达到提升效率的一种编程思想。

当然，在这个过程中会遇到很多麻烦，比如会碰上死锁的问题，同时去争抢同一个资源的问题等等。

为了解决这个问题，iOS里是使用加锁这个方法避免，当然，使用锁，会在一定的程度上影响执行的效率，我们需要在效率和安全之间寻找一个平衡点。

发布于 2016-09-09

9 2 条评论 分享 收藏 感谢 ... 收起



包荣瑄
市场营销

5 人赞同了该回答

在某一时间片只有一个线程在执行，由于系统为每个线程划分的时间片很小，所以宏观上看上去好像是多个线程在同时运行。

打个比方单线程运行十个qq多线程二十，线程多就是多一个大脑，就像人多一只手一样同事做两件事可以。

发布于 2016-10-19

5 1 条评论 分享 收藏 感谢 ...



周兆熊
软件工程师

9 人赞同了该回答

“单线程”程序

要想理解“多线程”，那么就要先从“单线程”说起。

大家都知道工厂“流水线”作业，里面的工序是一环扣一环的，只有前面的一道工序完成之后，才能够启动下一道工序。这其实和“单线程”的原理非常的相似。

在“单线程”里面，程序的功能是顺序执行的，只有前面的流程都成功执行之后，后面的流程才能够被执行到。

“多线程”程序

“多线程”，顾名思义，就是多个“单线程”，每个线程独立地完成相关的功能。

“多线程”的优点

“多线程”在大型软件程序中有着很广泛的应用，其优点如下：

第一，将原来在一个大流程中实现的功能放到了多个小流程中，程序更加的简洁和易于阅读。

第二，将不同的功能放到不同的线程中，提高了程序的执行效率。

第三，“多线程”使得程序的模块化更强，有利于追踪程序执行过程和排查问题。

总结

“多线程”和“单线程”分别对应“并行”和“串行”，是软件开发人员必须要掌握的一种程序设计的方法。设计合理的“多线程”程序不仅逻辑清晰、易于阅读，而且程序的执行效率高，对于软件产品效率和质量的提升具有很重要的意义。

请参考：[程序中的“多线程”](#)

[程序中的算法](#)

发布于 2015-03-09

9 添加评论 分享 收藏 感谢 ...

相关 Live 推荐

刘看山 · 知乎指南 · 知乎协
侵权举报 · 违法和不良信息
联系我们 © 2017 知乎

相关问题

更多

7 个回答被折叠（为什么？）



王森

java运算符 编辑话题经验

使用匿名身份回答

B I | H “ </> ☰ ☷

☹ ▶ Σ

写回答...

相关 Live 推荐

刘看山 · 知乎指南 · 知乎协
侵权举报 · 违法和不良信息
联系我们 © 2017 知乎