<<并行程序设计导论>>

图书基本信息

书名：<<并行程序设计导论>>

13位ISBN编号：9787111392842

10位ISBN编号：7111392841

出版时间：2012-12-1

出版时间：机械工业出版社华章公司

作者：Peter Pacheco

页数：252

译者：邓倩妮

<<并行程序设计导论>>

内容概要

《并行程序设计导论/计算机科学丛书》编著者Peter S.Pacheco。

本书全面涵盖了并行软件和硬件的方方面面，深入浅出地介绍如何使用MPI（分布式内存编程）、Pthreads和OpenMP（共享内存编程）编写高效的并行程序。
各章节包含了难易程度不同的编程习题。
本书可以用做计算机科学专业低年级本科生的专业课程的教材，也可以作为软件开发人员学习并行程序设计的专业参考书。

<<并行程序设计导论>>

作者简介

作者：（美国）帕切克（Peter S.Pacheco） 邓倩妮 Peter S.Pacheco 拥有佛罗里达州立大学数学专业博士学位。
曾担任旧金山大学计算机系主任，目前是旧金山大学数学系主任。
近20年来，一直为本科生和研究生讲授并行计算课程。

<<并行程序设计导论>>

书籍目录

章节摘录

版权页： 插图： 接下来，我们仔细研究最后两个参数，由pthread_create生成并运行的函数应该有一个类似于下面函数的原型：void*.thread_function（void* arga_p）；因为类型voi d*可以转换为C语言中任意指针类型，所以args_P可以指向一个列表，该列表包含一个或多个thread_function函数需要的数值。
类似地，thread_function返回的值也可以是一个包含一个或多个值的列表。
在我们的代码中，调用pthread_create函数时，传入最后一个参数采用了一个常用的技巧：为每一个线程赋予了唯一的int型参数rank，表示线程的编号。
首先，我们先解释一下这么做的理由，然后再具体探讨如何做。
考虑以下问题：运行一个生成了两个线程的Pthreads程序，当其中一个线程遇到了错误时，我们或者用户如何才能知道是哪个线程出了问题呢？
我们不能简单地输出pthread_t对象，因为它是不透明的。
如果我们启动线程时赋予第一个线程编号为0，第二个线程编号为1，那么通过错误信息中线程的编号就能非常容易地判断是哪个线程出错了。
既然线程函数可以接收voi d*类型的参数，我们就可以在main函数中为每个线程分配一个int类型的整数，并为这些整数赋予不同的数值。
当启动线程时，把指向该int型参数的指针传递给pthread_create函数。
然而，程序员会用类型转换来处理此问题：不是在main函数中生成int型的进程号，而是把循环变量thread转化为voi d*类型，然后在线程函数hello中，把这个参数的类型转换为long型（第33行）。
类型转换的结果是“系统定义”的，但大多数C编译器允许这么做。
不过，如果指针类型的大小和表示进程编号的整数类型不同，在编译时就会收到警告。
在我们使用的机器上，指针类型是64位，而int型是32位，为了避免警告，我们用long型替代了int型。
需要注意的是，我们为每一个线程分配不同的编号只是为了方便使用。
事实上，pthread_create创建线程时没有要求必须传递线程号，也没有要求必须要分配线程号给一个线程。
还需要注意的是，并非由于技术上的原因而规定每个线程都要运行同样的函数。
一个线程运行hello函数的同时，另一个线程可以运行goodbye函数。
但与编写MPI程序的方法类似，Pthreads程序也采用“单程序，多数据”的并行模式，即每个线程都执行同样的线程函数，但可以在线程内用条件转移来获得不同线程有不同功能的效果。
4.2.4运行线程 运行main函数的线程一般称为主线程。
所以，在线程启动后，会打印一句：Hello from the main thread 同时，调用pthread_create所生成的线程也在运行。
这些线程通过第33行的类型转换代码获得各自的编号，然后打印各自的消息。
注意，当线程结束时，由于它的函数的类型有一个返回值，那么线程就应该返回一个值。
在本例中，线程没有需要特别返回的值，所以只返回NULL。
在Pthreads中，程序员不直接控制线程在哪个核上运行。
在pthread_create函数中，没有参数用于指定在哪个核上运行线程。
线程的调度是由操作系统来控制的。
在负载很重的系统上，所有线程可能都运行在同一个核上。
事实上，如果线程个数大于核的个数，就会出现多个线程运行在一个核上。
当然，如果某个核处于空闲状态，操作系统就会将一个新线程分配给这个核。

<<并行程序设计导论>>

编辑推荐

《并行程序设计导论》是一本精心撰写的、全面介绍并行计算的书籍。
作者循序渐进地展示了如何利用MPI、Pthreads和OpenMP开发高效的并行程序，教给那些专业知识有限、没有并行化经验的读者如何开发、调试分布式内存和共享内存的程序，以及对程序进行性能评估。

名人推荐

毫无疑问，随着多核处理器和云计算系统的广泛应用，并行计算不再是计算世界中被束之高阁的偏门领域。
并行性已经成为有效利用资源的首要因素。
Peter Pachec0撰写的这本新教材对于初学者了解并行计算的艺术和实践很有帮助。
——Duncan Buell，南卡罗来纳大学计算机科学与工程系 本书阐述了两个越来越重要的领域：使用Plhreads和OpenMP进行共享内存编程。
以及使用MPl进行分布式内存编程。
更重要的是。
通过指出可能出现的性能错误，强调好的编程实现的重要性。
这些主题包含在计算机科学、物理学和数学等多个学科中。
各章包含了大量不同难易程度的编程习题。
对于希望学习并行编程技巧、更新知识面的学生或专业人士来说，本书是一本理想的参考书。
——Leigh Little，纽约州立大学布罗科波特学院计算机科学系

<<并行程序设计导论>>

版权说明