

<<浮点计算编程原理、实现与应用>>

图书基本信息

书名：<<浮点计算编程原理、实现与应用>>

13位ISBN编号：9787111243830

10位ISBN编号：7111243838

出版时间：2008-8

出版时间：机械工业出版社

作者：刘纯根

页数：301

字数：456000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<浮点计算编程原理、实现与应用>>

### 内容概要

本书介绍了基本计算算法的实现和代码分析，主要内容有：C++与通用编程、大整数运算、超高精度的定点和浮点仿真库、x87 FPU编程、反汇编vC6浮点库和常见浮点编程技巧。其中，超高精度的定点和浮点仿真库是作者为进行科学计算而开发的，具有较大的参考价值。

## 作者简介

刘纯根，1995年9月毕业于中国人民解放军国防科学技术大学自动控制系，获自动控制工学学士学位，1998年4月毕业于同一院校航天动力学专业，获工学硕士学位。后进入中国酒泉卫星发射中心技术部工作，主要从事导弹航天测试发射总体技术研究，涉及弹道与轨道力学仿真、计算编程、

## &lt;&lt;浮点计算编程原理、实现与应用&gt;&gt;

## 书籍目录

出版说明前言	第1章 引论	1.1 计算有什么用?			
	1.1.1 基础科学	1.1.2 应用科学	1.1.3 工程项目	1.1.4 日常生活	1.2
超高精度计算有什么用?	1.3 计算编程概述	1.4 一些缩写的解释	第1部分 原理与实现:通用仿真库	第2章 代码概述	
	2.1 基本内容	2.2 使用C++?			
	2.2.1 C还是C++?				
	2.2.2 c++与通用编程	2.3 C++代码的设计	2.3.1 使用模板	2.3.2 操作符重载	
	2.3.3 选择接口函数	2.3.4 参数传递	2.3.5 返回值处理	2.3.6 计算异常与	
诊断信息	2.3.7 内存布局	2.4 计算代码的测试	2.4.1 随机输入测试	2.4.2 特	
殊值测试	2.4.3 恒等式测试	2.5 代码的使用	2.6 伪码	第3章 通用整型运算	3.1
基本概念	3.1.1 记数法与数制	3.1.2 整型编码	3.1.3 编码位数换算	3.2 通	
用整型编码	3.2.1 数据定义	3.2.2 JYPE的选取	3.3 通用整型四则运算	3.3.1	
加法	3.3.2 减法	3.3.3 乘法	3.3.4 除法	3.4 优化	3.4.1 加法
	3.4.2 减法	3.4.3 乘法	3.4.4 除法	3.5 符号处理	3.5.1 原码,还是补码?
	3.5.2 有符号通用整型的表示	3.5.3 符号操作	3.6 输入输出函数	3.6.1 输入	
函数	3.6.2 输出函数	3.7 代码使用	第4章 通用定点运算	4.1 基本概念	4.1.1
记数法	4.1.2 误差	4.1.3 舍入	4.1.4 有效数字	4.2 通用定点数编码	
4.2.1 数据定义	4.2.2 符号处理	4.3 四则运算	4.3.1 加法和减法	4.3.2 乘	
法	4.3.3 除法	4.4 辅助操作	4.5 代码使用:计算派	第5章 浮点数与IEEE浮点标准	
第6章 通用浮点运算	第7章 通用基本函数库第2部分	应用:x87FPU编程与VC6浮点库	第8章		
x87FPU编程	第9章 编写自己的浮点库	第10章 如何反汇编代码	第11章 VC6浮点函数库		
第12章 异常处理机制	第13章 浮点编程中的常见技巧	附录参考文献			

## 章节摘录

第1部分 原理与实现：通用仿真库第2章 代码概述2.3.6 计算异常与诊断信息在计算过程中，有时会出现一些事件使计算无法进行下去或者计算错误，例如在除法中遇到被零除、计算结果超出表示范围等，这些事件被称为计算异常（注意，这里的异常仅指溢出之类与计算有关的异常，不是指非法指针访问之类的系统异常）。

显然，一个稳定可靠的，计算系统必须处理计算异常，而如何处理计算异常的关键问题是你以什么观点看待它们：（1）一般很少发生，即使发生了，代码的用户（更高层次的开发员）应该知道并处理它。

随这种观点而来的自然是最简单的处理方式，即忽略它。

（2）是一种严重的错误，计算需要立刻停止以进行错误处理。

在c++中，抛出异常是非常适合这种观点的处理方式。

（3）可能造成错误，但没有严重到需要停止计算的地步。

当异常发生时，计算应当继续进行，但返回结果的同时也发出警告。

方式（1）是最常见也最容易实现的方式，但它给后续的开发带来不便。

高层开发员可能需要编写计算异常的检测代码。

通常，检测代码的层次越高，效率就越低，编程越困难。

方式（2）过于敏感，如果某些用户并不关心计算异常，没有编写相应的计算异常处理代码，那么这个抛出的异常将破坏程序的流程，通常会导致程序终止。

另外，编写异常处理代码也会提高计算代码在组织或阅读上的难度。

方式（3）比较折中，而且在需要时，可以方便地转换为方式（1）或方式（2）。

例如，如果你不关心，那么简单地忽略警告即可；如果你极其关切，那么可以在得到警告时再抛出异常。

本书代码就使用了方式（3），而这意味着除了返回一个值之外，还需要设置一些诊断信息供用户查询，以保证用户可以检查计算结果。

信息均存在维护的问题。

系统提供诊断信息的目的，是为了在出错时提示用户错误所在，而什么时候进行检测则是用户决定的，系统无法确定，故一般系统要保留相关信息以待用户提取，直到有新信息产生时，才会明确清除旧信息。

这种特性使得诊断信息的维护有点困难。

如果用户得到了诊断信息，却不知道信息是当前产生的还是以前遗留的，那么这些信息除了使他困惑之外还能有什么用处呢？有两种维护方式可以避免这种局面：一是在每个接口函数中进行维护，即在每个接口函数的入口处明确清除以前遗留的信息。

这样处理以后，用户每次得到的信息总是最新的（Win32 API就是这样处理的）。

## <<浮点计算编程原理、实现与应用>>

### 编辑推荐

《浮点计算编程原理、实现与应用》主要介绍基本计算算法的实现和代码分析，涉及任意长的整数计算、任意范围和精度的实数计算(包括定点运算和浮点运算)、常见数学函数的实现和Visual C++6.0(简称VC6)浮点库C形式的反汇编代码分析。

全书分两个部分：第一部分讲述计算编程的基本原理与算法。

第二部分讲述Intel公司的x87 FPU与Microsoft公司的VC6浮点库。

最后简要介绍了x87指令集和《浮点计算编程原理、实现与应用》的源码。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>