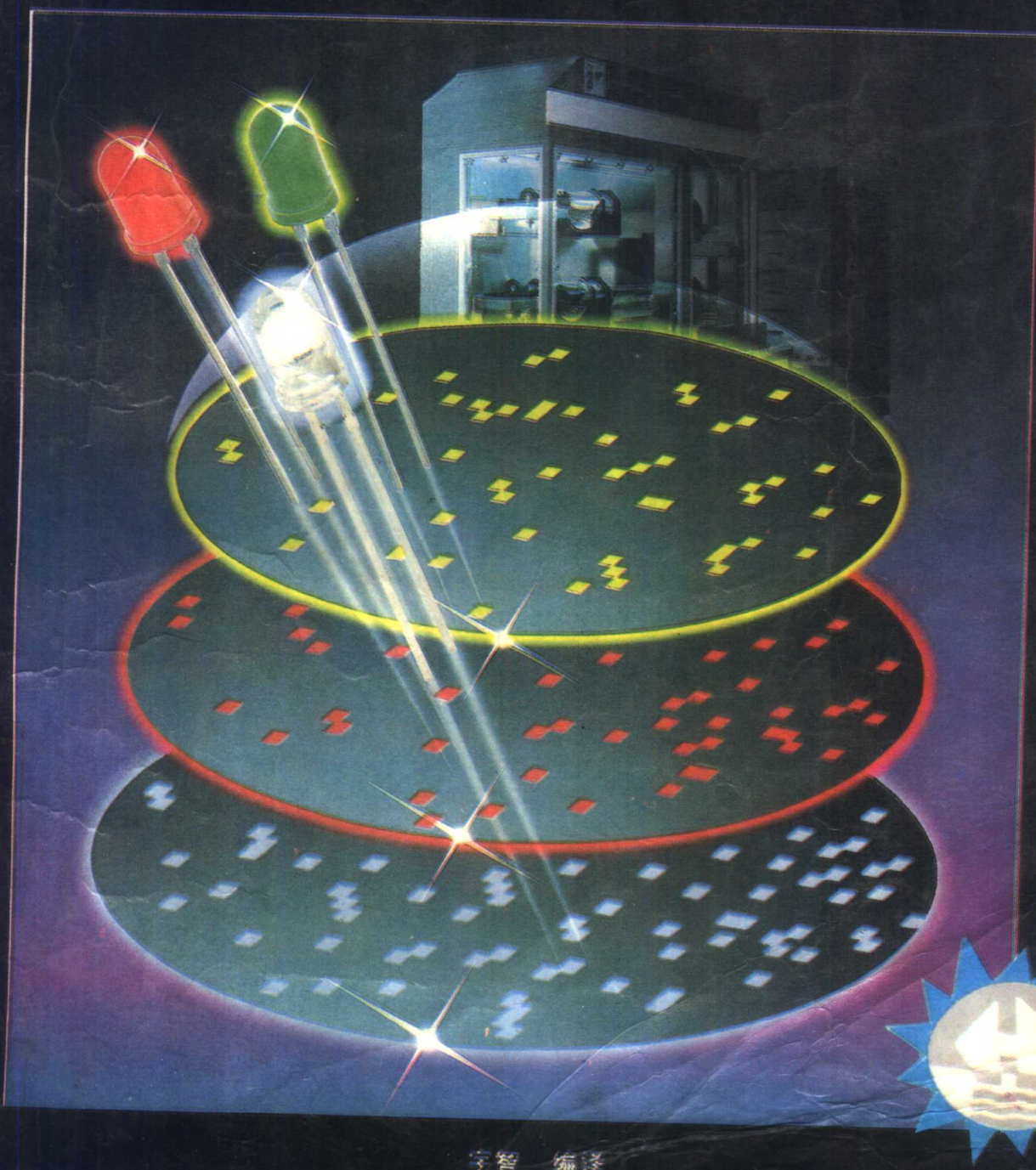


适用于 IBMPC XT AT 286 386 及其兼容机

HOPE

Microsoft C

科学与工程工具库



北京希望电脑公司



宇智 编译

882862

Microsoft C

科学与工程工具库

宁智等 编译

- 科学与工程常用数据分析
- 科学与工程二维三维图形
- 支持多种打印机和绘图仪
- 适用于 MS C 5.0 以上版

北京希望电脑公司

一九九一年九月

版 权 所 有
翻 印 必 究

■ 北京市新闻出版局

准印证号： 3313—90313

■ 订购单位： 北京8721信箱资料部

■ 邮 码： 100080

■ 电 话： 2562329

■ 传 真： 01—2561057

■ 乘 车： 320、332、302路
车至海淀黄庄下车

■ 办公地点： 希望公司大楼一楼
往里走 101 房间

前 言

Microsoft C 科学与工程工具库是一个适用于 IBM PC、XT、AT 及其兼容机的通用 C 函数库,这些函数解决了科学与工程应用领域中常见的数据分析和图形问题。工具库中的所有函数均用 Microsoft C 编写,它们包含在十七个不同文件中,分别解决了下面十七类问题:

数理统计	矩阵运算
多重回归	数据平滑
曲线拟合	复变函数
数值积分与微分	线性规划
富里叶分析	非线性方程求解
Lotus 文件转换	特殊函数
实型联立方程	二维图形
复型联立方程	三维图形
异步通信	

除了解决科学与工程数据分析问题之外,本工具库中还提供了功能强大的图形子程序,可以在各种 CRT 图形适配器上绘制基于科学数据的各种二维、三维图形,包括各种坐标图形、曲线图、条形图、圆饼图、分布图和等高线等,支持比例变换、平移变换、旋转变换、移动转换、裁剪、面消隐、线消隐和视面控制。工具库中给出了用汇编语言编写的异步通讯支持例程,为各种针式打印、HP 系列激光打印机和多种标准绘图仪提供了接口程序,因此,所有图形均可打印输出。

本书详细介绍了每个函数,包括其功能、调用序列、调用示例、参数说明、特别说明和示例程序,并给出了编译链接命令行和示例程序的运行结果。本书最后列出了全部源程序,这些源程序可以直接引用,也可以根据特殊要求加以修改。

译者于北京
一九九一年九月

目 录

第一章 导论

1.1 用 C 科学与工程工具库的启动	3
1.2 Quick C 用户应特别注意	4

第二章 C 语言的数据结构

2.1 C 语言数组	5
2.2 C 中的复数表示	8
2.3 使用双精度浮点值	9

第三章 概要统计

函数 SummaryStats	11
-----------------------	----

第四章 联立方程

函数 GaussJordan	13
函数 ComplexGaussJordan	14

第五章 多重回归

函数 MultipleReg	16
----------------------	----

第六章 曲线拟合

函数 PolyCurveFit	18
函数 CubicSplines	19
函数 CalcSpline	20

第七章 数值积分与微分

函数 IntegrateVector	22
函数 IntegrateFunction	22
函数 RungKutta	23

第八章 Fourier 分析

函数 FFTCalc	26
函数 FFTInvCalc	27
函数 WindowFFTDData	28
函数 FFT2DCalc	29
函数 PowerSpectrumCalc	30

第九章 Lotus 文件转换

函数 LotusFileSave	32
------------------------	----

第十章 矩阵运算——实数和复数

函数 MatProd	34
函数 CMatProd	34
函数 MatScalarProd	35
函数 CMatScalarProd	36
函数 MatAdd	37

函数 CMatAdd	38
函数 MatTranspose	38
函数 CMatTranspose	39
函数 MatDeter	40
函数 MatInvert	40
函数 CMatInvert	41
函数 CyclicJacobi	42
第十一章 数据平滑	
函数 DataSmoothSg	44
函数 DataSmoothWeights	45
第十二章 复数函数	
函数 ComplexMath	47
函数 CExp	47
函数 CMag	48
函数 CAngle	48
第十三章 科学与工程图形	
13.1 引论	50
函数 InitSEGraphics	51
函数 CloseSEGraphics	52
函数 DefGraphWindow	52
函数 SetPercentWindow	54
函数 SetCurrentWindow	55
函数 SetWin2PlotRatio	56
函数 ClearWindow	56
函数 ClearGraph	57
函数 BorderCurrentWindow	57
函数 SetViewBackground	58
函数 SetPlotBackground	58
函数 SetAxesType	59
函数 ScalePlotArea	60
函数 SetXYIntercepts	60
函数 DrawXAxis	61
函数 DrawYAxis	62
函数 LabelXAxis	64
函数 LabelXAxisWithStrings	64
函数 LabelYAxis	65
函数 TitleXAxis	66
函数 TitleYAxis	66
函数 TitleWindow	67

函数 DrawGridX	67
函数 DrawGridY	68
函数 DrawGrid	68
函数 AutoAxes	69
函数 LinePlotData	70
函数 BargraphData	70
函数 ScatterPlotData	71
函数 GroupPlotData	72
函数 PlotErrorBars	74
函数 PieChart	75
函数 ContourPlot	76
函数 ContourPlotLegends	79
函数 LabelGraphWindow	80
函数 LabelPlotArea	81
函数 StringLegends	81
函数 RealLegends	82
函数 ScreenDump	83
函数 PlotterOn	85
函数 PlotterOff	86
函数 CRTGraphOn	86
函数 CRTGraphOff	87
函数 SortDataX,SortDataY	87
函数 FindMinMax	88
函数 ConvertNum	88
函数 SelectColor*	89
函数 SetTextStyleXX	90
函数 SetLineStyleXX	90
函数 SetFillStyleXX	90

第十四章 三维绘图例程

14.1 引言	92
14.2 使用 C 语言实现 3 维图形库	92
函数 Init3D	93
函数 Close3DGraphics	94
函数 SetWorldCoordinates	94
函数 Ident	95
函数 Ident3	95
函数 Concat	96
函数 Concat3	96
函数 tlnit	97

函数 tInit3	97
函数 Xfrm2P	97
函数 Xfrm3P	98
函数 WorldTran2	98
函数 WorldTran3	99
函数 WorldRotate2	99
函数 WorldRotate3	100
函数 WorldScale2	100
函数 WorldScale3	101
函数 Persp	101
函数 Line3Abs	102
函数 Line2Abs	102
函数 Line3Rel	103
函数 Line2Rel	103
函数 Move3Abs	104
函数 Move2Abs	104
函数 Move3Rel	104
函数 Move2Rel	105
函数 Label3D	105
函数 PolyFill3D	106
函数 ScreenDump	107
函数 PlotterOn	107
函数 TurnPlotterOff	107
函数 CRTGraphOn	108
函数 CRTGraphOff	108
14.3 各种 Worldr 及 Plot3d 函数	
函数 SelectColor	108
函数 SetTextStyleXX	108
函数 SetLineStyleXX	109
函数 SetFillStyleXX	109
第十五章 线性规划	
函数 Simplex	110
第十六章 方程求根	
函数 BrentRoots	113
函数 NewtonRoots	114
第十七章 特殊函数	
函数 Tan, Gosh, Sinh, Sech, Arctanh	116
函数 Gamma, LogGamma	116
函数 IncGamma	117

函数 IncGammaComp	118
函数 Beta	118
函数 IncBeta	119
函数 Bessel	119
函数 ModBesselI	120
函数 ModBesselK	121
函数 ErrFunc	121
函数 ErrFuncCom	122
函数 ErrFuncR	122
函数 ErrFuncI	123
函数 Hermite	123
函数 Legend	124
函数 Laguerre	124
函数 Jacobi	125
函数 Tcheb	125
函数 DisplayErrorMessage	126
第十八章 异步通信	
函数 open_com	127
函数 send_com	127
函数 writeln_com	128
第十九章 错误信息	
19.1 常见的连接和运行错误	129
第二十章 源程序清单	
ASYNCXX. C 的源程序清单	131
ASYNCXX. H 的源程序清单	137
BARDEM. C 的源程序清单	137
BIGC. BAT 的源程序清单	138
CGJ. C 的源程序清单	139
CGJ. H 的源程序清单	140
CGJDEMO. C 的源程序清单	140
CMMATHDE. C 的源程序清单	143
COMPLEX. C 的源程序清单	145
COMPLEX. H 的源程序清单	147
CONTOUR. C 的源程序清单	148
CROSS2D. C 的源程序清单	149
CURVEDEM. C 的源程序清单	151
CURVEFIT. C 的源程序清单	154
CURVEFIT. H 的源程序清单	157
ERRORDEM. C 的源程序清单	158
FFT. C 的源程序清单	159
FFT. H 的源程序清单	167
FFT2DEMO. C 的源程序清单	167
FFTDemo. C 的源程序清单	168
FFTPower. C 的源程序清单	170
FFTWINDE. C 的源程序清单	172
FIR. C 的源程序清单	174
FUNCPlot. C 的源程序清单	177
GJ. C 的源程序清单	179
GJ. H 的源程序清单	181
GJDEMO. C 的源程序清单	181
GRAPHFFT. C 的源程序清单	183
GROUPDEM. C 的源程序清单	185
HIDELINE. C 的源程序清单	188

HOUSE3D. C 的源程序清单	191	ROOTS. C 的源程序清单	252
HPPLOT. C 的源程序清单	193	ROOTS. H 的源程序清单	257
HPPLOT. H 的源程序清单	202	ROOTSDEM. C 的源程序清单	258
INTEGRAT. C 的源程序清单	203	ROTAXES. C 的源程序清单	258
INTEGRAT. H 的源程序清单	208	SCATDEM. C 的源程序清单	259
INTEGRDE. C 的源程序清单	208	SCRNDUMP. C 的源程序清单	261
LINEDEM. C 的源程序清单	209	SEGRAPH. C 的源程序清单	263
LOGDEM. C 的源程序清单	210	SEGRAPH. H 的源程序清单	308
LOTUS. C 的源程序清单	213	SIMPLEX. C 的源程序清单	310
LOTUS. H 的源程序清单	215	SIMPLEX. H 的源程序清单	312
LOTUSDEM. C 的源程序清单	215	SIMPLEXD. C 的源程序清单	312
MATMATH. C 的源程序清单	216	SMOOTH. C 的源程序清单	313
MATMATH. H 的源程序清单	227	SMOOTH. H 的源程序清单	320
MATMATHD. C 的源程序清单	228	SMOOTHDE. C 的源程序清单	320
MISCIO. C 的源程序清单	230	SPECDEMO. C 的源程序清单	322
MISCIO. H 的源程序清单	232	SPECFUNC. C 的源程序清单	328
MULREG. C 的源程序清单	232	SPECFUNC. H 的源程序清单	345
MULREG. H 的源程序清单	235	33 <u>SPLINEDE. C</u> 的源程序清单	346
PIEDEM. C 的源程序清单	235	STATSDEM. C 的源程序清单	347
PLOT3D. C 的源程序清单	237	SUMSTATS. C 的源程序清单	348
PLOT3D. H 的源程序清单	244	SUMSTATS. H 的源程序清单	350
PLOTTERD. C 的源程序清单	245	TERMINAL. C 的源程序清单	351
REGDEM. C 的源程序清单	249	WORLDDR. C 的源程序清单	352
READ. ME 的源程序清单	250	WORLDDR. H 的源程序清单	382
RKFDEMO. C 的源程序清单	251		

第一章 导 论

C 语言科学与工程工具库是一个运行于 IBM PC、XT、AT 和大多数 IBM 兼容机的通用 C 函数集,这些函数解决了科学与工程应用领域中的常见的数据分析和图形问题,它们分别包含在 17 个不同的文件——每个文件包含了相关的函数。其分类见下表:

分 类	函数文件
一般统计	sumstats. c
多重回归	mulreg. c
曲线拟合	curvefit. c
数值积分与微分	integrat. c
Fourier 分析	fft. c
Lotus 文件转换	lotus. c
联立方程	gj. c
复数联立方程	cgj. c
矩阵运算	matmath. c
数据平滑	smooth. c
复数运算	complex. c
绘制图形	segraph. c
三维图形	plot3d. c
线性规划	lsimplex. c
非线性方程求解	roots. c
特殊函数	specfunc. c
异步通信	miscio. c

1. 数理统计:

计算数据集的概要统计,包括最小值、最大值、数据值范围、方差、标准偏差、平均误差和众数。

示例程序:statsdem. c

2. 多重回归:

对数据集进行最小二乘线性多重回归。函数结果并返回回归方程系数、y 的估值、残差、估值的标准误差、回归系数的标准误差、测定系数(R 平方值)和相关系数(R 值)。

示例程序:regdem. c

3. 曲线拟合:

我们提供两种不同的拟合技术,将实验数据拟合成光滑曲线。第一种是多项式曲线拟合,第二种是三维样条曲线拟合。

示例程序:curvedem. c 和 splinede. c

4. 数值积分:

根据实验数据计算出由实验数据曲线所包围的面积,或一个任意函数曲线所包围的面积。采用的算法是 Simpson 1/3 法则和 Simpson 3/8 法则。

示例程序: integrde. c

5. 微分方程求解:

解决涉及一阶微分方程的初值问题。使用变步长的 Runge—Kutta—Fehlberg 方法。

示例程序: rkfdemo. c

6. Fourier 分析:

计算数据集的快速富里叶正变换和快速富里叶反变换;使用快速富里叶技术(FFT)计算 2 维 Fourier 变换;并解决在调和分析中所使用的矩形、Parzen、Hanning、Welch、Hamming 和 Exact Blackman 窗口;计算波形的能谱周期图。

示例程序: fftdemo. c, fft2demo. c, fftwinde. c, fftpower. c, graphfft. c

7. Lotus 1—2—3 文件转换:

以某种格式把数值数组保存到磁盘上,这些数值数组可以使用 Lotus 文件输入选择项被输入到一个 Lotus 1—2—3 电子表中。表示列和行头的字母数字字符串也可被存放,使得这些字符串能被作为“标号”输入到 Lotus 1—2—3 中。

示例程序: lotusdem. c

8. 联立方程组的求解:

用 Gauss—Jordan 法求解一个线性方程组。函数返回系统解矢量、X 变量系数矩阵的逆阵和 X 变量系数矩阵的行列式。书中给出了二种不同的函数,一种用于求解实型联立方程组,另一种用于求解复型联立方程组。

示例程序: gjdemo. c, cgjdemo. c

9. 矩阵运算:

矩阵运算模块包含了计算矩阵乘积、矩阵求和、矩阵转置、矩阵行列式、逆矩阵和求实对称矩阵特征值的函数。函数可用于处理实数和复数。

示例程序: matmathd. c, cmmathde. c

10. 复数运算:

计算复数的加、减、乘和除,以及复数的指数、复数的模、复数的极角等。

示例程序: cgjdemo. c

11. 数据平滑和卷积

使用 Savitzky—Golay 函数来减少实验数据集中的噪声量。广义卷积函数还允许用户设置加权系数和正规因子。

示例程序: smoothdem. c

12. 绘制图形:

通过使用本工具库中的特殊例程,可以产生直线图、条形图、散布图、等高线堆栈式直线图和栈式条形图、条形图组、Linear、Log 和半 Log 比例图。CRT 图形适配器支持包括 CGA、EGA、VGA、Hercules 单显和其它设备(阅读“编译程序手册”)。打印机支持包括 Epson MX、FX 和 LQ 在内的打印机、HP Laser Jet 系列、HP Think Jet 和 Toshiba P1351、P351。

示例程序: linedem. c, bardem. c, scatdem. c, groupdem. c, logdem. c, errordem. c, contour. c, curvedem. c, graphfft. c, fftwinde. c, smoothwe. c, smoothsg. c

13. 3 维绘图例程:

使用 2 维视图投影来处理 3 维物体的建模和视图的通用系统。物体在显示前可以在三

维坐标系中进行比例变换、旋转变换、平移变换,此外,还包括平行和透视投影变换等。

示例程序:house3d.c,rotaxes.c,cross2d.fir,funcplot.c,hideline.c

14. 线性规划:

求解线性规划方程的标准单纯形表系统。

示例程序:simplexd.c

15. 非线性方程求根:

采用二分法、Erents 和 Newton 法求解非线性方程的根。

示例程序:rootsdem.c

16. 特殊函数:

科学与工程领域中出现的许多数学方程不包括在标准数学函数库中。这些特殊函数包括 gamma、beta 和贝塞尔函数,还包括实数和虚数的错误函数、双曲三角函数及正交多项式的求解方法。

示例程序:specdemo.c

17. 异步通信:

Microsoft C 对 RS-232 通信没有足够的支持。在 asyncxx 库中的函数包括基本的 RS-232 支持。程序员可设置所有由通信参数支持的 IBM DOS。

示例程序:plotterd.c

§ 1.1 用 C 科学与工程工具启动

在你的硬盘上创建一个新目录:

MKDIR\MC006

并将主磁盘的内容拷到上面,改变工作目录到:

C:\MC006

为了运行一个特殊的示例程序,用它调用的任何库文件(.obj 或 .lib)编译链接主源文件。下面表示如何做:

cl gjdemo.c gj.c miscio.c

如果你正在使用 Quick C 编辑程序,你可设置程序表到正确的 *.mak 文件。在这个例子中,gjdemo.mak 是正确的 *.mak 文件。

主源文件与它直接和间隔调用的文件一起被编译并链接,产生一个可执行程序。

curvedem.c 示例程序包括下列文件:

cl curvedem.c curvefit.c mulreg.c gj.c miscio.c segraph.c woridrr.c asyncxx.c hpplot.c

curvedem.c 程序调用在 curvefit 文件中的库函数 PolyCurveFit,而 PolyCurveFit 调用 mulreg 文件中的多重回归例程,多重回归例程再调用 gj 文件中的 Gauss-Jordan 例程,最后用科学与工程图形库绘制一个给定的 curvefit 结果。

curvedem.c 主程序有如下的程序头:

```
/* Curvedem Program */  
# include <math.h>  
# include <graph.h>  
# include "curvefit.h"
```

```
# include "miscio. h"  
# include "segraph. h"
```

在主程序中调用的所有函数需要通过 `#include <*.h>` 或 `"*.h"` 文件被调用,这样对于从其他被单独编译的模块中输入的函数调用就产生了正确的代码。

文件 `miscio. c` 也包括在标准 Microsoft C 库中不包括的函数,这些函数包括文本窗口的屏幕控制指令如 `CoToxy` 和 `ClrScr` 函数,也包括常量 `pi(π)` 和 `pi/2` 的定义。

有关在 Microsoft C 中编译链接程序的详细内容可参阅 Microsoft C 手册。

§ 1.2 Quick C 用户应特别注意

如果你正使用任何图形并使用 Quick C 1.5,则需要一个较大的堆栈。通过选择 Run Runtime 的选择项 `stack` 可调整堆栈。设置栈空间为 16K 左右。如果你正使用 Quick C 2.0,则可通过选择 Options Make Compiler 博 Large 可使用一个大的内存。款你已对每个编译程序均使用 `Rebulid All`,使用合适的堆栈大小/呐存模型可编译全部源文件。若不能 `Rebuild All` 源文件则可能产生一个链接错误。

在警告级上使用 Quick C 编辑程序可测试全部库函数和示例程序。如果你在大于 1 的警告级上编译和链接这包中的任何函数,在代码中一定行上你可能会看到信息数据转换。因为 Microsoft 将自动转换变量类型,如从整型到实型,所以未包括所有类型转换。

如果你在 `miscio. c` 库中使用 Quick C 编辑程序,你将注意到 `ClrScr` 和 `ClrEol` 函数不工作。这是因为 Quick C 编辑程序截断了由 `ClrScr` 和 `ClrEol` 函数产生的 `bios` 调用。

学习如何使用这些函数的最好方法是学习你的科学和工程磁盘提供的例子程序。编译这些示例程序的最简单的方法是装载在 Quick C 编辑程序中与之相关的程序 (`Linedem. mak`, `bardem. mak`, `scatdem. mak`, `groupdem. mak` 和 `logdem. mak`) 然后编译这个程序。

第二章 C 语言的数据结构

§ 2.1 C 语言数组

Microsoft C 语言科学与工程工具库中的所有函数均使用指向实数数组的指针作为函数的参数。与本工具库的 Pascal 和 Modula 模块不同, C 语言工具库中没有一个由所有库公用的预定义的向量和矩阵数据类型标准集, 相反, 所有的数组被作为数的线性开向量处理。

带一个一维数组作为参数的函数的编码形式如下:

```
FFTCalc(float * x, float * y, int n);
```

对该例程的调用形式类似于:

```
float a[100];
```

```
float b[100];
```

```
int n;
```

```
n=100;
```

```
FFTCalc(a, b, n);
```

传递数组名与传递数组的起始地址相同, 不需要任何下标括号。

主调用程序可把任意大小(最大为 64K 字节)的数组传递给用这种方式编码的例程。参数 n 可由被调用的例程使用, 以确定所传递的数组的大小。如果 n 的值超过了数组 x 和 y 的实际大小, 则无法通知被调用的例程。与 Pascal 和 Modula 语言不同, C 不对数组进行下标检查。如果一个数组下标越界, 则被调用的例程会覆盖程序中的其它部分, 从而破坏数据并且可能使程序崩溃。

带一个 2 维数组作为参数的函数的编码形式与上例相似:

```
MatAddr(float * x, float * y, float * z, int r, int c);
```

参数 r 和 c 通知例程正在被传递的数组维数为 $r \times c$ 。数组 x, y 和 z 维数必须与这个值完全相同, 否则过程会错误地计算数组下标。

如果一个 2 维数组名传递给了一个要求一维数组或指针的例程, 则 Microsoft C 编译程序将产生警告信息。因为编译程序的这种反常现象, 所有的 2 维数组都应通过传递 2 维数组的第一个元素的地址而被传递。

下例给出了设置 2 维数组, 并把它们传递给 C 函数的错误和正确方法:

错误方法:

```
{  
float x[2][2];  
float y[2][2];  
float z[2][2];  
:  
MatAdd(x, y, z, 2, 2); /* 不能仅用名来传递二维数组 */  
}
```

正确方法:

```

float x[2][2];
float y[2][2];
float z[2][2];
;
MatAdd(&x[0][0], &y[0][0], &z[0][0], 2, 2);

```

二维数组的另一个例子将产生一个不正确的结果:

错误方法:

```

{
float x[10][10];
float y[5][5];
float z[6][6];
;
MatAdd(&x[0][0], &y[0][0], &z[0][0], 2, 2);
/* x, y, z 的维数不是 2×2 */
}

```

这个例程不会产生预期的结果, 因为三个数组的大小是不相同的, 同样, 被传递到函数 MatAdd(2, 2) 中的维数与源和目标数组的维数不完全相同。正确调用 MatAdd 函数的例子如下:

正确方法:

```

{
float x[2][2];
float y[2][2];
float z[2][2];
;
MatAdd(&x[0][0], &y[0][0], &z[0][0], 2, 2);
}

```

这个方法有优点, 也有缺点。

优点是被调用的函数使用了尽可能少的数据空间, 它仅利用了运行这个函数所需的最小内存空间, 所有临时工作数组在堆上分配, 以使堆栈空间最小。

缺点是要求二维数组必须与保存数据所需的内存空间一样大, 许多程序员习惯于设置大的工作数组(如 10×10 , 100×100), 这些大数组能处理他们可能遇到的最大数组。当分析较小数组(2×2 , 4×8)时, 它们正好位于较大数组内部。当一个 2×2 矩阵的数据被放在一个 10×10 大小的数组中时, 结果是产生了很多未被利用的内存。

程序员如何来写一个能处理各种大小的数组的程序呢? 有二种方法。第一种是专为一维数组编码, 一旦知道了数组的维数, 就可在堆上分配保存数组所需的准确内存空间。例如, 创建一个足够大的浮点数组, 该数组可由用户输入维数。

```
float *x; /* x 是指向浮点数的指针 */
```

```
int r,c;
```

```
scanf("%d %d",&r,&c);
```

```
x=(float *)calloc(r*c,4);    /* 在堆中分配 r*c*4 个字节,并将此数组的起始  
                               地址赋给 x. */
```

如果矢量是双精度(8字节/实型)而不是浮点型(4字节/实型),则 calloc 函数将以如下方式被调用:

```
x=(float *)calloc(r*c,8);
```

对用 2 维坐标指定的任何元素,计算数组的相对偏移地址的方法非常简单。在上例中,如果数组大小是 3×5 ($r=3$ 和 $c=5$),则以下面形式存取元素 $x[2][3]$:

```
x[2*c+3]=3.2425;
```

其中 $c=5$ 。

用这种方式处理数组可以使用本程序包中的所有例程。

编一个能处理各种 2 维数组程序的另一种方法是使用一个大的 2 维工作数组,例如, 50×50 。通常,数组中只能被部分赋值,并且这些数据在数组的左上角。你可以写一个 C 函数,该函数能从工作数组中把感兴趣的数据(0~2 列,0~2 行)移到一个一维行向量中,工作数组中不用的行和列不拷贝进该一维向量,然后该一维向量被作为数据传递到分析例程中,一旦数据被返回,可以再将这些数据移回工作数组中。

例如,用户想转换一个 3×3 数组,使用下面的简单程序就可把值输入到工作数组中:

```
float a[100];    /* 用于向函数传递参数的线性向量 */
```

```
float b[10][10]; /* 二维工作数组 */
```

```
int i,j,r,c;
```

```
scanf("input # of rows and columns %d %d\n",&r,&c);
```

```
for(i=0;i<=r-1;i++)
```

```
    for(j=0;j<=c-1;j++){
```

```
        printf("Input Value for Row # %d Column # %d\n",i,j);
```

```
        scanf("%f",&b[i][j]);
```

```
    }
```

在内存中, 3×3 数组存贮在 10×10 工作数组中的实际情况如下,