

中国科学院希望高级电脑技术公司高级程序设计丛书

Quick BASIC 4.0

高级结构化程序设计技术

(美) Hergl 著

秦笃烈 刘学宗 毛佳飞

郭德成 阎岩 译

秦笃烈 审校



海洋出版社

中国科学院希望高级电脑技术公司高级程序设计丛书

Quick BASIC4.0

高级结构化程序设计技术

[美] D.Hergert 著

秦笃烈 刘学宗 阎 岩 译
毛佳飞 郭德成
秦笃烈 审校

海洋出版社

内 容 提 要

本书内容对Microsofs Quick BASIC 4.0的功能、用户接口环境、开发工具以及图形等方面的重大改进作了全面介绍,可以说是一本非常实用而新颖的书。除程序设计技巧以外,书中还给出了六个完整程序,包括:LIST(排序和文件列表的数据类型的数据结构)、TWENTY-ONE(游戏)、SURVEY(如何有效使用顺序文件的信息搜集和分析程序)、EMPLOYEE(数据库管理程序)、QUICK CHART(建立图表的程序)、ADVANCED MENU(菜单建立程序)。

Quick BASIC 4.0

高级结构化程序设计技术

[美] D.Hergert 著

秦笃烈 刘学宗

编译 毛佳飞 郭德成

阎 岩

审校 秦笃烈

特约编辑 鞠玉兰 秦人华

海洋出版社出版(北京市复兴门外大街1号)

新华书店北京发行所发行 东升印刷厂印刷

开本: 787×1092 1/16 印张: 18.75 字数: 500千字

1990年9月第一版 1990年9月第一次印刷

印数: 1—3000册

*

ISBN 7-5027-1196-1/TP·18

定价¥: 13.50元

序 言

BASIC语言是国内外最流行的计算机语言。计算机科学在迅速发展，BASIC语言也在发展。1987年美国国家标准局正式公布BASIC语言国家标准。BASIC语言的发明人始终致力于该语言的自身完善。目前，在微型机上已形成三个BASIC语言新版本：True BASIC，Quick BASIC以及Turbo BASIC。从美国国家标准来看True BASIC更为接近，但Quick BASIC和Turbo BASIC也不无自己独到之处。本书主要参考Microsoft出版社1988年出版的Hergert Douglas所著QuickBASIC（第二版）编译。众所周知，目前出书难的情况影响了读者对计算机科学最新成就的了解和吸取。在这种情况下中国科学院希望高级电脑技术公司支持我们出了一套程序设计丛书。它们是最新TurboBASIC实用大全，‘Quick BASIC高级软件开发工具包’，‘C语言程序设计技巧与开发’以及本书。我们认为这些书都是写得很出色的。主要内容均取自美国最新佳作。对希望公司这一举动以及海洋出版社的合作我谨向他们表示衷心感谢。我们希望这些书籍对国内读者能有所裨益。对其中心不足之处热望读者及时赐告。本书程序盘委托希望公司发行也可以直接与作者联系。

秦笃烈

1990.6.3.于首都医学院

地址：右安门外首都医学院BME系 100054

目 录

第一部分 Quick BASIC4.0简介

序 言

第一章 理解Quick BASIC环境

- 1.1 Quick BASIC编程环境 (1)
- 1.2 Quick BASIC语言 (5)
- 1.3 Quick BASIC菜单的用法 (6)
- 1.4 结论 (22)

第二章 子程序和函数的用法

- 2.1 Quick BASIC4.0的子程序 (22)
- 2.2 Quick BASIC4.0的函数..... (30)
- 2.3 一个示范程序 (33)
- 2.4 结论 (37)

第三章 编制程序

- 3.1 函数Menu%——在屏幕上提供菜单 (38)
- 3.2 把过程文件结合到程序中 (44)
- 3.3 多个程序模块的用法 (46)
- 3.4 如何建立Quick程序库 (47)
- 3.5 结论 (48)

第二部分 语言论题和示范程序

第四章 数据类型和数据结构：磁盘目录检查程序List

- 4.1 程序List简介 (50)
- 4.2 基本数据类型 (51)
- 4.3 Quick BASIC的算术运算 (53)
- 4.4 运算的优先级 (54)
- 4.5 内装数值函数 (55)
- 4.6 字符串数据类型 (55)
- 4.7 字符串函数 (56)
- 4.8 逻辑值 (69)
- 4.9 逻辑值的内部表示 (60)
- 4.10 符号常量..... (61)
- 4.11 Quick BASIC支持的数据结构..... (62)
- 4.12 程序List的运行 (67)
- 4.13 程序List内幕 (70)

4.14	程序List中的数据类型和数据结构.....	(72)
4.15	程序List中对按时间顺序值的处理	(76)
4.16	结论	(81)
4.17	程序LIST清单.....	(82)
第五章 判别与循环: 游戏程序Twenty-One		
5.1	程序Twenty-One简介.....	(100)
5.2	QuickBASIC4.0的判别结构	(101)
5.3	QuickBASIC4.0的循环结构.....	(105)
5.4	游戏Twenty_One的玩法	(108)
5.5	程序Twenty_One内幕	(115)
5.6	结论.....	(130)
5.7	程序Twenty_One清单	(131)
第六章 顺序数据文件: 用于调查的程序Survey		
6.1	程序Survey简介.....	(145)
6.2	QuickBASIC处理顺序文件的方法.....	(146)
6.3	程序Survey的用法.....	(152)
6.4	程序Survey内幕.....	(161)
6.5	结论	(170)
6.6	程序Survey清单	(171)
第七章 随机存取数据文件: 数据库管理程序Employee		
7.1	QuickBASIC处理随机存取文件的方法	(188)
7.2	程序Employee运行示范	(193)
7.3	程序Employee内幕	(198)
7.4	结论	(208)
7.5	程序Employee清单	(208)
第八章 图形: 实用作图程序Quick Chart		
8.1	程序Quick Chart简介.....	(225)
8.2	Quick BASIC中的图形	(227)
8.3	程序Quick Chart的使用方法.....	(233)
8.4	程序Quick Chart内幕.....	(239)
8.5	结论	(255)
8.6	程序Quick Chart清单	(255)
第九章 事件捕捉: 高级菜单程序 AdvMenu		
9.1	事件捕捉.....	(279)
9.2	程序 AdvMenu	(282)
9.3	程序AdvMenu内幕	(283)
9.4	结论.....	(285)
9.5	程序AdvMenu清单	(285)

第一部分 Quick BASIC 4.0简介

第一章 理解Quick BASIC环境

Microsoft Quick BASIC 4.0版是一种新的编程环境, 它把Microsoft 最新编译技术的速度和效率与解释程序的直接性和方便性结合了起来。在本章中我们将探索这一新的环境, 并揭示它是如何使程序展开的步骤变得更为合理的。我们还将概括地看一看新的 Quick BASIC语言某些主要的特点。

在本章后面我们将检查产生, 编辑, 存储, 执行与调试Quick BASIC程序的种种工具。我们会看到菜单驱动的使用者接口是如何工作的, 并且将列出它呈现的特点。本章虽不能替代 Quick BASIC文献介绍材料, 但这里提供的概述将在很大的范围内展示 Quick BASIC程序包的面貌。

1.1 Quick BASIC编程环境

首先, 在MS-DOS提示符下键入QB命令, 开始一个与Quick BASIC的会话。

C>QB

这时, Quick BASIC 的菜单行就出现在屏幕的顶端, 光标闪烁表明编辑程序已经准备好接受新程序的第一个语句。现在, 已处于Quick BASIC编程环境中了。

这一环境的实际细节, 即菜单命令, 显示屏幕, 键盘与鼠标器等既条理清楚而又直观。无论编程背景是在BASIC解释程序中还是在Quick BASIC的先前版本, 这些细节都是容易控制的。开始编制新程序之前, 应该先了解Quick BASIC 的最重要的新的设计特点:

- “灵巧”的编辑程序
- 上下文有关帮助
- 窗口
- 多模块程序
- 新调试设备
- 独立的编译后的程序 (EXE文件)
- Quick程序库

下面各节简要叙述这些特点。

1.1.1 灵巧的编辑程序

Quick BASIC “灵巧”的编辑程序远不止是进入与修改程序文本的一个工具。这个编辑程序被设计的目的在于进入每一程序行时能迅即并几乎察觉不到地执行一系列重要任务:

首先, 编辑程序检查程序行的句法。如果命令有句法错误, QuickBASIC就会识别出问题并在屏幕上显示一个信息框表明出了什么错误。

这个编辑程序还保证每一新行代码与QuickBASIC标准行格式相一致。按这一格式, 保留字以大写形式出现, 所有操作码和操作数都用空格隔开。可以以任何方便的方式进入每一程序行, 编辑程序自动地重组该行以与标准格式相匹配。

灵巧的编辑程序是 Quick BASIC 的新环境的入口点, 这个环境与合理的展开过程一道, 提供了更高的执行速度。这一编辑程序还附有其它特点, 它的大多数基本键盘命令与 WordStar 的控制字符是相同的, 而许多程序设计人员对后者都很了解。另外, 这个编辑程序中装置了一个灵敏的联机求助设备, 我们在下一节对其进行讨论。

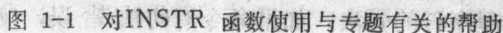
在Quick BASIC 中有好几种方法可以获得帮助,与专题有关的帮助是其中最有用

1. 在编辑程序中键入一个QuickBASIC保留字。

这样, QuickBASIC 就会显示一块叙述所输入的命令或函数的用法与句法的求助版面。

$$\text{Space pos \%} = \text{INSTR}$$

图1-1表示产生的Quick BASIC显示。阅读这个简明摘要之后,大致用不着再去查阅文档就可继续编写程序行了(否则,为了解更多的细节和范例,恐怕总要查阅Microsoft Quick BASIC Language Reference一书)。要清除求助版面继续工作,请按Esc键。



窗口是QuickBASIC环境的另一重要特点。开始与QuickBASIC会话时,屏幕上显示

2

视见窗的下方是一个小窗口，叫作直接窗口。在把语句编入程序之前可以在直接窗口对 Quick BASIC命令与函数进行实验。可在屏幕上开启的第三扇窗口叫作观察窗。这个窗口显示调试会话时有关程序的信息。下面几小节简要叙述各类窗口的用法。

•View (视见) 窗口

典型地，一个长的Quick BASIC程序含有许多小的过程，它们每一个都负责完成一项具体任务。程序开头部分常称为主程序部分，它通过调用下面的过程控制着整个程序的作用。为了易于了解具有这种一般结构的程序的展开，Quick BASIC把过程个别地显示在屏幕的窗口。只要程序驻在内存中，就可以在View窗口中检查具体的过程。

从磁盘初次装入一个程序时，其主程序部分就出现在编辑程序的View窗里。为了观看这个程序的任何一个过程，可直接下拉View菜单并选择SUB命令。这个命令会按字母顺序排列出程序中的所有过程。例如，图1-2表示了一个名为Schedule的程序，其过程表如下。

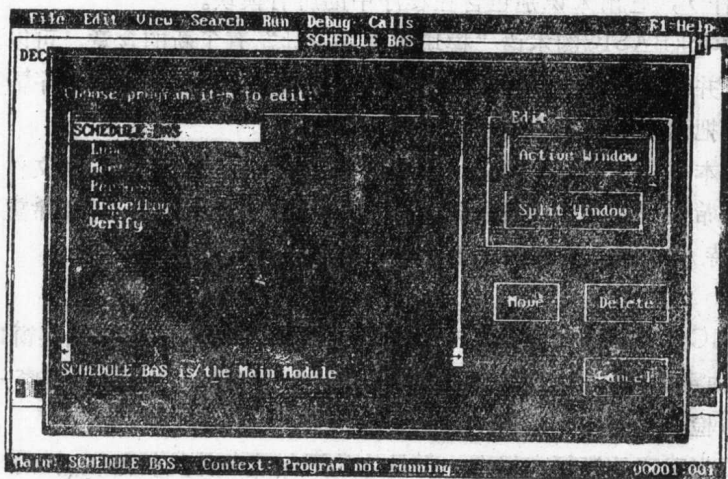


图 1-2 观察一个用SUB命令列出的过程表

Schedule程序共有5个过程，可以选择其中任何一个放到窗口中观察与编辑。

View菜单还有一个命令可把屏幕分为两个隔开的View窗。用这两个View窗，可以同时看两个不同的过程。在哪一个窗口进行编辑都行。包含编辑光标的窗口是工作窗口，按F6功能键可以激活屏幕上的另一窗口。

•Immediate (直接) 窗口

直接窗口是 Quick BASIC 新的特点，它保持了BASICA解释程序的一个便利之处，即有解释直接方式命令并马上在屏幕上显示结果的能力。直接窗口可用来对Quick BASIC进行快速的实验。

几乎可以把任何可执行的Quick BASIC命令放入该窗口。其响应是，Quick BASIC把显示转移到输出屏幕并显示出该命令的结果（按任何键可返回编辑程序）。这是在程序中使用某命令之前了解该命令如何作用的有用方法。

直接窗口在调试Quick BASIC程序时也有着重要的作用。在本章的后面部分我们将对其进行讨论。

•Watch (观察) 窗口

仅在调试会话过程中观察窗口才出现在屏幕上,其目的是给出关于程序执行情况的具体信息,尤其是选择的变量或表达式的值的变化情况。用Debug 调试菜单中的命令确定要显示什么信息。在程序运行的同时检查观察窗口中发生的变化,常可以发现程序逻辑上的问题。否则,这些问题是难以测定的。

1.1.4 多模块程序

Quick BASIC4.0 版的一个主要革新就是增加了对多模块程序的支持。在开始编写Quick BASIC应用程序时,几乎总是当然地展开在其它编程问题中常用的具有普遍目的的过程。Quick BASIC 4.0鼓励把这样的过程组为一个个文件分别存入磁盘。然后,在准备写一个新程序时,可以把任意号码的这样的文件作为程序模块装入内存。

一个程序可以容纳所有要放入内存的各个模块文件。所有模块中的过程随时可以调用。存储程序时,每个模块仍存回磁盘上的自己的文件,而Quick BASIC也产生一个特殊的文件(叫作MAK文件),它用来识别包含在程序中的所有模块。

这一特点使QuickBASIC的术语“模块化程序设计”有了新的含意。可以展开及试验具有普遍目的的过程组并以任意文件号把它们的源码存入磁盘。然后,在新程序需要这些试验过的例行程序时,可把这些文件作为模块直接装入内存。

为简单明了,本书第二部分中的应用程序都被安排为自包含的单模块文件。在运用这些文件时,我们可能希望把几个公共过程分开成为其本身的模块文件,可能希望充分利用QuickBASIC支持模块的优越性。在第三章,我们将讨论模块的使用。

1.1.5 调试

Quick BASIC编程环境中建立了一个强有力的调试工具系统。与观察窗口一道,Quick BASIC提供了种种跟踪方式,它们一行接一行或一个过程接一个过程地贯穿一个程序。利用这些方式,可以检验程序中个别命令及语句的结果。

还可以设置预先确定的停顿点以在一个具体的语句或条件处暂停程序的运行。停止运行时,可检视输出屏幕,利用立即窗观察变量及表达式的值。甚至可以在程序中作一些更改,然后再恢复程序的运行。

调试菜单使得Quick BASIC的调试特点很容易进入。在运用几个特殊的键盘命令控制调试菜单时,发现与修正错误都比以往的方法有效得多,即使在又长又复杂的程序中也是如此。

1.1.6 产生EXE文件

EXE文件是一个能在MS-DOS直接执行的编译程序。Quick BASIC4.0提供一个简单的过程以产生这样的文件。一旦在Quick BASIC环境展开并检验了一个程序,则仅使用一个菜单命令就可在磁盘上产生一个该程序的编译版本。Quick BASIC以机内方式履行这个过程的所有步骤。要在Quick BASIC编程环境外部执行这个结果程序,只要在MS-DOS提示符下键入文件的基名。

例如,假使已经使用这个过程编译了一个名叫Survey的程序。QuickBASIC把其编译代码以文件名SURVEY.EXE.存在磁盘上。要执行这个程序,就键入程序名称作为一个MS-DOS命令:

```
C>SURVEY
```

在本章后面,我们还要考察产生EXE文件的步骤。

1.1.7 Quick程序库

Quick程序库是一个编译过程的集成。与EXE文件一样, Quick程序库文件也可从Quick BASIC环境直接产生。可以从任意模块或模块形成Quick程序库。

要使用Quick程序库的过程,可在启动Quick BASIC时把文件装入内存。QB命令的/L选择项把Quick程序库与Quick BASIC编程环境一道装入。例如,下面的命令装入了名为TOOLBOX.QLB的程序库:

```
C>QB/L TOOLBOX.QLB
```

实际上,这样一个程序库的过程就相当于扩展了语言。利用程序就可调用这些过程,甚至程序库过程源码没有被包含该程序中也行。在第三章我们将更细致地讨论Quick程序库。

和新编程环境一样, Quick BASIC也有一些主要的新的成份。Quick BASIC4.0与可能已经用过的其它版本普遍兼容,包括BASICA解释程序及编译程序的以前版本。所以,大多数现存程序无需修改或很少修改就可以执行。然而, Quick BASIC4.0也包含着一些重要的命令与函数,这些在以前是不能用的。在本章的下一节,我们将列出Quick BASIC语言的一些新内容。

1.2 QuickBASIC语言

利用Quick BASIC4.0几个新语言特点的优越性,可写出效能高,作用大并且结构精巧的程序。程序产生,调试和修改起来也比以往更为容易。

这里要讨论的重要的新语言成份是:

- 子程序与函数
- 循环
- 扩展的数据类型与数据结构
- 结构化循环与判定
- 调用别种语言子程序

这些题目的大多数将在第二部分就其应用实例被详细讨论。下面几节对这些有意义的新的语言特点作一简单的预习。

1.2.1 子程序与函数

Quick BASIC的设计鼓励把一个长的编程问题分解为小的过程。Quick BASIC支持两种过程:子程序与函数。这两个组织单元有几个重要的共同特点:

1. 调用任一过程都能把自变量值传到该过程中。
2. 以机内方式,过程内部使用的变量是局部的,也就是其值防止了外部干扰并且对过程外的变量没有作用。
3. 通过参数或通过值都可把自变量传递到一个过程,这一区别决定了是否把值送回到该自变量名称下的调用子程序。

存储一个被分为子程序与函数的程序时, Quick BASIC在程序顶端自动地产生特殊的过程说明语句。这些语句以保留字DECLARE开头,它们允许Quick BASIC检查送入一个过程调用的自变量的类型与数量。

在第二章,我们将研讨所有这些特点,并检验子程序,函数与DECLARE的具体句

法。

1.2.2 循环

Quick BASIC4.0支持循环过程。循环就是在执行某项任务的过程中一次或多次地调用一个过程。几个共用算法采用了这项技术的优点。其中一个典型的例子是叫作Quick sort的分类例行程序。在第四章我们将研究一个快速分类的版本。

1.2.3 数据类型与数据结构

QuickBASIC4.0的最有用，最有意思的扩展包括了下面这些新数据类型与数据结构。

- 长整数。它是一个值的范围比标准整数更宽的数字类型。
- 固定长度的字符串变量。它是一个表示长度不变的文章值的字符串类型。
- 记录变量。它是一个混合的数据结构，可表示不同类型的多个值（记录变量属于相应的用户定义类型）。
- 符号常量。它是任何字符串类型或数字类型的命名的常数。

数据类型与数据结构是第4章的主要论题。

1.2.4 结构判定与循环

Quick BASIC支持两种有力的判定结构：

- 块结构化的IF语句。它表达了不同作用过程中的条件选择。
- SELECT CASE语句。它在一系列CASE块内构造替换作用。

这两个结构的基本不同在于选择程序要执行的作用的方法上。在IF语句中，判定基于一个或多个表达式。相比之下，SELECT CASE语句却是在目标检验值与CASE值表之间进行比较。

至于重复结构，Quick BASIC除更为传统的FOR...NEXT循环之外，还支持多方面的DO循环。DO循环重复执行编码块，条件表达式用来决定循环范围。使用这一结构，可产生DO...WHILE及DO...UNTIL循环。

第5章将集中讨论判定结构与循环。

1.2.5 调用别种语言过程

Quick BASIC4.0支持各语言间的调用。利用这一特点，可把任何一种Microsoft语言写成的过程并入一个Quick BASIC程序。适合的语言包括Microsoft C, Microsoft Quick C, Microsoft Pascal, Microsoft FORTRAN与Microsoft Macro Assembler (MASM)。为把一个例行程序由别种语言引入Quick BASIC程序，首先要编译并联结这例行程序到Quick程序库中，然后再把程序库装入Quick BASIC环境。有关语言间调用的更详细的情况，请见Quick BASIC文献。

1.3 Quick BASIC菜单的用法

当开始与Quick BASIC的会话时，编辑程序就已经准备好，可以开始编程工作了。下面从编辑程序中产生一个程序的典型步骤：

1. 从键盘直接把新程序键入Quick BASIC编辑程序，或者从磁盘把一个现存程序装入编辑程序。
2. 使用各种编辑特点，对程序进行必要的改变。
3. 把程序的当前版本存入磁盘，可同时打印程序清单。

4. 执行程序

5. 如果结果与预想不同, 激活选择的调试特点并返回步骤 2。

6. 如果结果与预想相符, 再一次执行它。这一次把编译码作为一个可执行的 MS-DOS 程序存入磁盘文件。

图 1-3 中表明, Quick BASIC 环境为简单有效地完成这些步骤提供了所需的工具。从 Quick BASIC 菜单中选择一个命令就可进入使用这些开发工具。

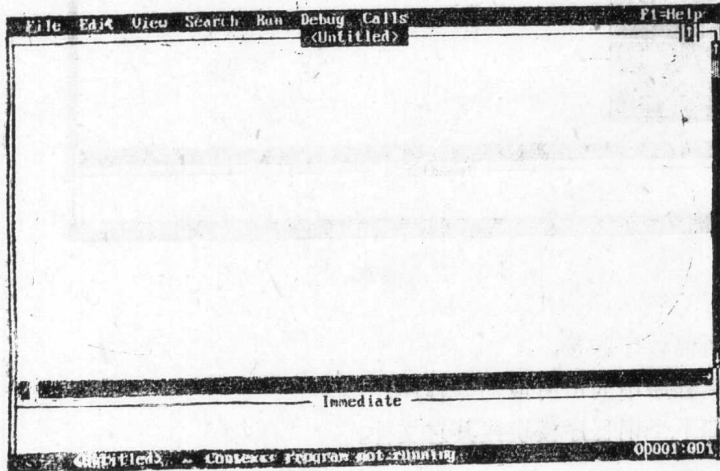


图 1-3 Quick BASIC 环境

要“下拉”一菜单且显示命令选择项变量, 按下 Alt 键, 同时按要检视的菜单的头一个字母。例如, 按 Alt 键及 F (Alt-F) 就拉下 File 菜单。要从当前显示的菜单中选择一个具体的命令, 按该命令的大写字母。有省略号 (...) 跟在菜单命令名后面意味着, 选择该命令时, 对话框显示在屏幕上。

某些菜单命令也有特殊的简便的键组合。可用它们从键盘直接选择命令, 而无须采用贯穿菜单的较多步骤。例如, 按 Shift-F5 就从 Run 菜单产生了 Start 命令。

Quick BASIC 还支持使用鼠标器, 它是控制 Quick BASIC 菜单的替换设备。利用鼠标器可以

- 下拉菜单及选择菜单选择项。
- 激活窗口并改变它们的大小。
- 把光标移到编辑程序内的新位置。
- 在编辑程序中选择文章的块。

鼠标器在调试中也有具体的应用。

要了解所有菜单的详细情况, 请查阅 Quick BASIC 文献。它给出了所有菜单命令的完备描述以及使用指导。下面几节的目标着眼于在每一菜单上 Quick BASIC 提供的最重要的工具。

1.3.1 File 菜单

可用图 1-4 中所示 File 菜单中的命令完成下列任务:

- 从内存清除当前程序。
- 从内存把程序装入编辑程序。

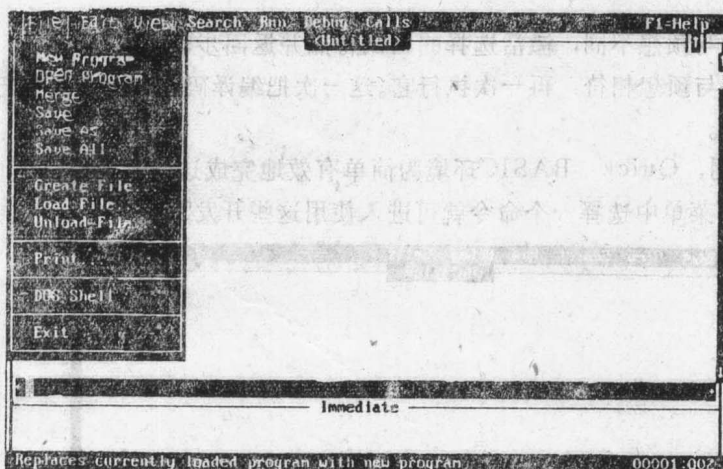


图 1-4 File菜单

- 把磁盘文件与当前程序联接。
- 存储当前在编辑程序中的所有或部分程序。
- 产生、装载或卸下个别程序模块或其它文件。
- 打印程序清单
- 产生到MS-DOS的暂时的或永久的出口。

下面几节简要探究File菜单中的每一命令。

1. 命令New program

New program命令从Quick BASIC的内存清除当前程序。只要在最后的存储操作之后对程序模块作出改动，Quick BASIC就在屏幕上提供一个对话框以及以下话：

One or more loaded files are not saved. Save them now?对话框有Yes, No和Cancel三个开关。

可以用键盘，也可以用鼠标器去答复对话框。例如，要在New program对话框中选择一个开关，可直接按Y或N，也可用鼠标器操纵开关。还可以用Tab键选择对话框的任何成份。如选好一个开关，就可以按Enter键或者空格键来执行这个开关的作用。Esc键在何时都可取消操作，并从屏幕清除对话框。

2. 命令Open program

Open program命令可从一个或多个磁盘文件中读出程序并将其装入编辑程序。这个命令的对话框包括一个File Name文本框，可把要打开的文件名称送入其中。该对话框还有一个目录框，它列出当前目录中的程序文件。要装入一个程序，可直接把其名称键入文本框并按Enter键。也可按照下列步骤从目录框选择一个程序：

- (1) 按Tab键选择目录框
- (2) 使用箭头键选择目标名或按该文件名的头一个字母。
- (3) 按Enter键装入该文件

要看另一目录或从另一目录装入文件，需把路径名键入文本框或在目录框中选择目录说明符。

如程序由一个以上模块构成,可使用Open program一次装入所有模块文件。为识别多模块程序中各个模块,Quick BASIC读出MAK文件(该文章文件列出了组成程序的所有模块)。送入文件名,Open program就寻找MAK文件。若MAK文件不存在,Quick BASIC就表示出程序仅有一个模块。

只要一个BASICA程序作为文本文件被存储在磁盘上,就可使用Open program把这个文件装入Quick BASIC编辑程序。要从BASICA中产生一个ASCII文本,可使用SAVE命令的A选择项。例如:

SAVE "SURVEY.BAS", A

若省略了A选择项,对BASICA存储的程序Quick BASIC编辑程序无法阅读其形式。

如果愿意,可以使用任何外部文本编辑程序或词处理机展开Quick BASIC程序,但要知道这是不带任何具体控制字符的简单文本文件,它对于Quick BASIC来说是无意义的。

3. 命令Merge

Merge命令可把一Quick BASIC编码文件联入当前程序。这一命令的对话框包括一个File Name文章框与一个列出当前目录的框。如内存中的程序包含一个以上的模块, Merge命令可把目标文件装入当前模块(即视见窗显示的模块)。我们将在第3章讨论Merge命令的用法。

4. 命令Save

File菜单共有三个存盘命令: Save, Save As与Save All

Save命令可把当前模块的最后版本存入磁盘。当文件至少已被存储过一次,而最后一次存储操作之后又想把改动之处也记录下来,典型地要使用这个命令。

头一次存储模块,要使用Save As命令。这个命令显示一个对话框,为该模块准备的文件名能送入其中(事实上,不管用Save还是用Save As命令,当对给出的模块进行第一次存储操作时,屏幕上总显示出对话框)。

用Save As命令存储程序时有两种存储格式可供选择:

(1) QuickBASIC- Fast Load and Save

(2) Text- Readable by Other Programs

其一是存储程序编码的Quick BASIC本身的格式。它使得在其后把程序读回到Quick BASIC时,装载操作快速有效。其二则产生一个简单文章文件,用这一格式可把程序源码转移到某个其它的编辑环境(可在MS-DOS使用TYPE命令观看以文章格式储存的文件内容)。

使用Save All命令可以把当前程序的所有模块存入磁盘。每个模块被存入一个分离的文件。对于多模块程序,这个命令还产生必要的MAK文件,它列出组成该程序的各模块文件的名称。MAK文件的基名称是主模块的名称。例如,假定使用Save All存储了一个名为Survey的多模块程序,其主模块为SURVEY.BAS,则模块清单就存于SURVEY.MAK名下。Quick BASIC对于单模块程序不产生MAK文件。

5. 命令File

使用Create File, Load File和Unload File命令可建立,装入和删除模块文件。此外,这三个命令可用于include文件及非程序文档文件(include文件包含有代码行,它们是在编译时由\$ INCLUDE语句读入程序的。第三章我们将讨论这个过程)。

Create File命令可对当前程序产生一个新模块。可在该命令的对话框中为新模块键入

名字。然后用View菜单中的SUB命令作用于这一模块。例如，可用SUB命令把一个过程从一个现存模块移到新模块。

Load File命令可从磁盘读出模块文件，并且把该模块装入当前程序。对话框包括当前目录列表和可把要装入的模块名称键入其内的文本框。同样地，View菜单中的SUB命令可用来观察新装入模块中的过程。

Unload File可从当前程序中删除一个模块。对话框列出程序中所有模块。当删除一个模块时，QuickBASIC从内存删除该模块并从程序的MAK文件中删除掉该模块的名称。但该模块未从磁盘中被删除

6. 命令Print

Print命令可把当前程序的全部或部分清单送至打印机。这个命令的对话框给出下面选择项，请见图1-5。

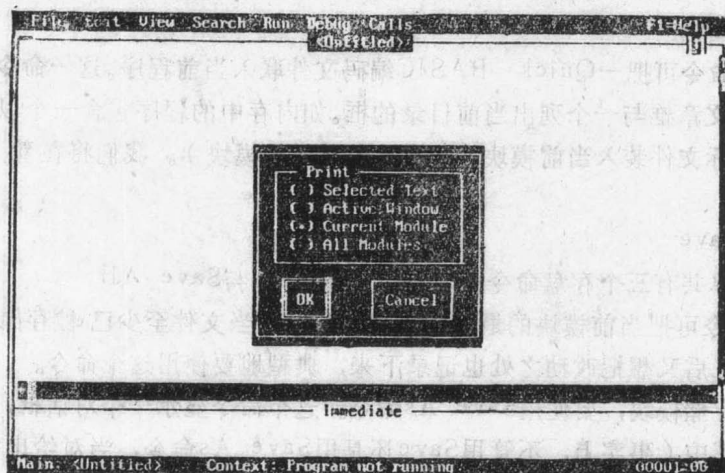


图 1-5 Print命令对话框

Selected Text把文本的选块送至打印机。Active Window选择项打印存于活动窗内的单过程。Current Module选择项打印当前程序模块的所有过程。若要打印完整的程序，包括所有模块，那就选取All Modules选择项。

由于Selected Text选择项作用于文章选块，则可通过选择编辑程序中的程序的一部分来开始这项操作。可用键盘或鼠标器选择文本。键盘方法如下：按下Shift键的同时按任何一个数字区上的方向键（例如，用Shift键和一下箭头键选择文本的行）。如用鼠标器选择文本，就在按下左鼠标器开关的同时移动鼠标器指示点掠过目标文本。该被选文本被反相显示。

7. 命令DOS Shell

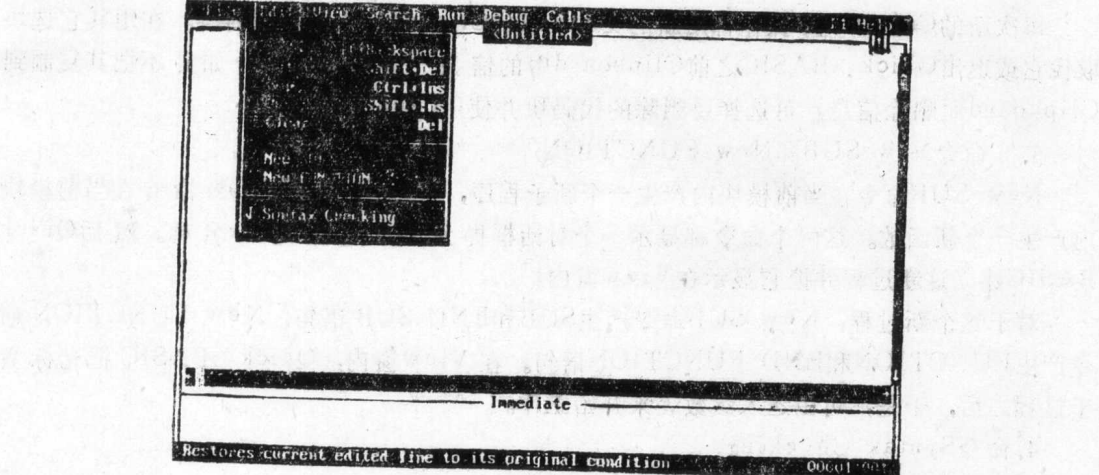
DOS Shell是一个很有用的命令，用它可暂时脱离Quick BASIC而去执行MS-DOS下的一个或多个操作。使用MS-DOS EXIT命令就会返回Quick BASIC；

C>EXIT

8. 命令Exit

File菜单上的Exit命令可结束与Quick BASIC的当前对话。在选用Exit时，如当前程序中含有任何未存的改动，Quick BASIC就显示一个对话框。该对话框给出选择项以

中的命令执行下列任一操作:



紅樓夢詩集 卷之四

- 将一个代码块从当前程序中的某位置移到另一位置（切割与粘贴）。
- 把代码块从某位置复制到另一位置（复制与粘贴）。
- 从程序中删除代码块。
- 产生一个新过程窗口。
- 开启或关闭Quick BASIC的句法检查

可用现成的简便的键来选择这头五个命令中的任何一个。

1. 命令Undo

Undo命令取消在当前程序行中作出的任何编辑改动,恢复该行的原内容。只有在光标仍处于需要恢复的程序行中时才可用这一命令。当光标已移到另一行时,Quick BASIC就记录了编辑改动,Undo命令就不可用了。

2. 命令Cut, Copy和Paste

Cut (切割) 与 Paste (粘贴) 命令一起把语句块从程序内部的一个位置移动到另一个位置。下面是Cut与Paste操作的步骤:

- (1) 选择要移动的代码块。使用键盘 (Shift键加数字区上的 方向键) 或鼠标器都可做此选择。
- (2) 选用Cut命令。Quick BASIC把被选出的代码从程序中删除 并把它存入名为缓存器(Clipboard)的内存单元。
- (3) 把光标移到要粘贴所选代码的位置上。
- (4) 选用Paste命令。Quick BASIC把被选代码从缓存器(Clipboard)粘贴到目标位置。

可以用Copy和Paste命令把语句块复制到新位置而不删除原代码块。下面是拷贝并粘贴操作的步骤:

- (1) 选择要复制的编码块。