

用QuickC编写DOS

实 用 程 序

王 玉 书 编 写

北京希望电脑公司

目 录

第一章 引言	(1)
1.1 对实践的简短说明	(2)
1.2 启动Quick C	(2)
1.3 使用Microsoft C	(5)
1.4 与Microsoft C链接	(6)
1.5 使用这些实用程序	(6)
第二章 目录实用程序	(7)
2.1 FILEATTR 列出目录里所有文件的属性	(7)
2.2 FILESIZE 列出目录里所有文件和它们的规模	(12)
2.3 FINDFILE 自动地在多个目录和网络找文件	(17)
2.4 LCD 快速改变目录	(24)
2.5 LISTDIR 列出所有驱动器和网络上的所有目录	(29)
2.6 MOVE 移动一些文件到新目录	(35)
2.7 RENDIR 目录改名	(40)
2.8 VTREE 目录树结构的直观表示	(43)
2.9 ZAP 去掉整个目录树结构	(50)
第三章 安全实用程序	(57)
3.1 ALTER 改变文件的属性字节	(57)
3.2 SECURE 加密文件	(64)
3.3 VAULT 用算法和保密字加密文件	(68)
3.4 WIPEFILE 完全粉碎磁盘或网络上的文件	(74)
第四章 性能实用程序	(79)
4.1 ATDATE 设置AT日历上的日期	(79)
4.2 ATTIME 设置AT日历上的时刻	(83)
4.3 CLEAN 删除所有目录和网络里的所有.bak文件	(88)
4.4 PSWAP 交换打印机接口	(92)
4.5 TIMEMARK 用于计时程序显示当前日期和时刻	(96)
4.6 TURBOKEY 提高键盘速度	(99)
4.7 VCOPY 提示确认的全局拷贝	(102)
4.8 VDEL 提示确认的全局删除	(109)
4.9 VMOVE 提示确认的全局移动	(114)

4.10 VWIPE	提示确认的全局粉碎	(120)
第五章	文件实用程序	(128)
5.1 COMPARE	比较两文件的不同	(128)
5.2 DBINFO	等价于dBASE “显示结构” 的命令	(133)
5.3 FILEAPP	连接两个文件	(140)
5.4 FILECASE	改变文本文件的字体	(145)
5.5 TEXTSRCH	在一些文件里找文本串	(150)
5.6 TOUCH	改变文件的日期和时刻	(154)
第六章	批文件实用程序	(161)
ASK	提示批文件的响应	(161)
第七章	打印机实用程序	(165)
7.1 LASER	设置HP激光打印机模式	(165)
7.2 PMODE	设置EPSON型打印机模式	(170)
7.3 PRINTLN	发送一行文本到打印机	(175)
第八章	字处理实用程序	(177)
8.1 DETAB	去掉文档文件和程序中的tab	(177)
8.2 WSASCH	转换Wordstar文件成文本	(183)
词汇表		(187)

第一章 引言

C最初由贝尔实验室雇员丹尼斯·里奇 (Dennis Ritch) 于1972年提出的。它在运行UNIX操作系统的DEC PDP-11上做了大部分研究工作。然而里奇不是以C为起点。这种语言很大程度受其他两种语言的影响: 理查德 (Martin Richards) 研制的BCPL和汤普森 (Ken Thompson) 研制的B语言。B语言也受BCPL的影响, 因此C的根源是演变而不是创建。

由于C语言在微机上的使用增进了这种语言的可接受性和有用性, 因此C语言迅速地流行起来。然而, 微机使用C也引起了C语言的混乱, 因为没有关于C的实际标准。在C出现一些年以后才有C语言关于UNIX的标准。而对于运行MS-DOS的微机需要研究新的标准。因而, 1983年美国国家标准研究所 (ANSI) 确立了建立新标准的任务。这个标准继续发展并致力于微机上C语言的实现, 以便减少含混性。

这些研究的成果就是在过去几年里C语言得到了微机程序设计员的大力支持。很明显C语言编译程序市场到处可见。使C语言流行起来的部分原因在于人们对C能力的理解, 它允许程序设计员把关键建造和剪裁他们自己的函数库。使用C的程序设计员能把大的任务化为较小的、更紧凑的和更易于管理的一些小部分。这种“化小”有利于在需要一些类似小程序的应用中构造可重用的构造块。

对C增进兴趣的另一个理由是程序设计员要记住的语句量不大。纯C仅有32个关键字。语句如此之少消除了含混性, 使之容易写出应用。由于C既不是高级语言也不是低级语言, 它仅定义不多的控制和数据处理语句。这对程序设计员有很大的好处, 因为程序小容易看懂。在C程序中几乎没有额外开销。在编译和运行时, 只有功能上必要的过程和函数被包括进去。

也许对于C的最大赞美声是它的可移植性。你能在一台机器上写程序, 而毫无困难地把它移到另一台机器上。这个特点对程序设计员有极大的吸引力, 因为计算机工业总是处在不断变化的状况。学习在新机器上研制软件要花大量的时间。对一项应用编码需要大量的时间和金钱。要用另一种语言写这个程序的话确实如此。由于用C写的软件可移植, 因此重新构思一个程序完成一项任务所必要耗费的时间和金钱极少。所以, 用C写程序的好处之一是既省时间又省钱。

对于用C编程越来越受欢迎有许多明显的理由。一些新的C编译程序使C更易使用, 并且使得用C编程成为更吸引人的编程方法。Microsoft的Quick C是这样一个编译程序, 它使C更流行。Quick C对C语言增加的一些性能使C成为更强有力的编程工具。

Quick C和市场上不太贵的其他编译程序使我确信, 用C编程具有一定的灵活性, 使用这种语言能快速有效地编写出许多能力非常强的程序。由本书的实用程序你会看到, 有许多类型的程序用非常短的代码写了出来。事实上, 大多数程序大约160行长, 其中还有一些行用于注解和结构。

1.1 对实践的简短说明

本书中的所有列出的都是激光打印机打印出的源码文件里的东西。可以担保这些程序能像本书所表明地那样工作。如果你发现某个程序功能有问题，可能你敲的源码与列出的不一样，或者你的Quick C编译程序配置不正确，或者你正在使用不向下兼容的后来版本的Quick C。

如果实验有问题，首先检查敲的东西。然后弄清楚在你的驱动器上正确地安装了函数库，也弄清楚通知Quick C找函数库的环境变量是活动的。如果还有问题，可能你要购买装有本书源码和已编译的程序的软盘。

本书中的实用程序是使用中（medium）存储模式编译的。其中许多在其他存储模式下也能工作。但是，由于它们是按中模式设计和测试的，这些程序仅推荐作为参考。由于Quick C隐含是中模式，因此你在Quick C环境敲ALT R和字母C能编译这些程序。ALT R使你能运行Run菜单，C选出这个编译选择项。

1.2 启动Quick C

若想启动Quick C和输入它的程序编译环境，在DOS命令提示符下敲QC，立刻会弹出图1所示的主屏幕。在屏幕的顶行有可见的水平提示菜单。从主屏幕可选出一个动作。选择项是File、Edit、View、Search、Run、Debug、Calls和Help。因为还没有输入程序，你可以选用File装入一个程序或建立一个新的程序文件。

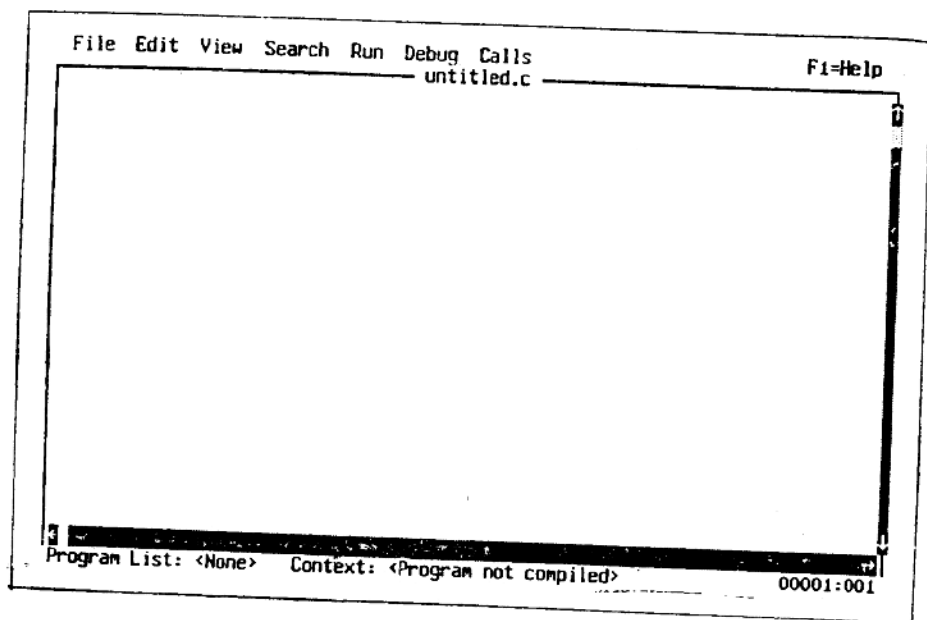


图1 主屏幕

屏幕各部分

Quick C是围绕View窗口（视察窗口）和信息区构造的。其他窗口在一定条件下

The screenshot shows the Turbo C++ IDE with the 'Debug' menu open. The menu options are: 'Add Watch...', 'Delete Last Watch Shift+F2', 'Delete All Watch', 'Trace On', 'Screen Swapping On', 'Toggle Breakpoint F9', and 'Clear All Breakpoints'. Below the menu, the main window displays the following C code:

```

#include <stdio.h>

Prototype:  int printf(const ch

Returns:    the number of chara

D

*/

main(int argc, char *argv[])
{
    printf("\nAA-Alter Attribs. QC Edition. (C) Copr 1987, Robert Alonso\n");
    numofargs = argc - 1;
    if (numofargs < 1)
        /* print help text */
        {
            printf("\nUsage: C:\>alter filename.ext [/r/h/s]\n");
        }
}

```

At the bottom of the screen, a black error dialog box is displayed with the text: 'Fatal Error C0002: (1 of 1)'. Below the error box, the status bar shows 'Program List: <None>' and 'Context: <Program not compiled>'. The address '00084:057' is visible in the bottom right corner.

View  

View窗口有几个重要部分。屏幕底处的线称做状态线。它通知编辑程序正在编辑文件的状态。这条状态线给出程序研制时有用的信息。它看起来如:

Error 窗口

• 3 •

下拉菜单

若你对Quick C环境已有了经验，会注意到通过左右移动光标和按Enter键可以访问屏幕顶处的命令。当按了Enter键，出现一个菜单，它是由屏幕的顶向下的。这种类型的菜单通常称作下拉式（pull-down）菜单。

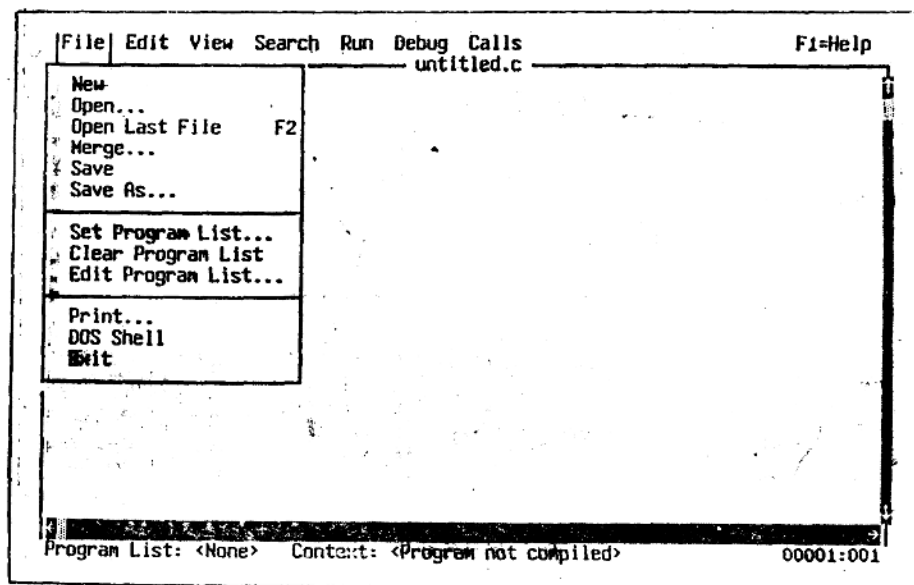


图3 File下拉菜单

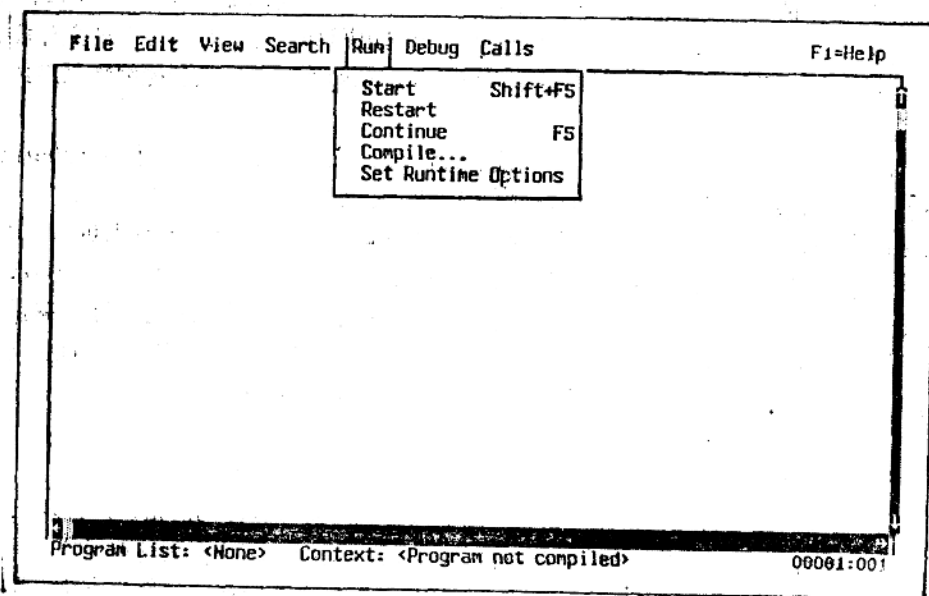


图4 Run下拉菜单

下拉式菜单变得日益流行起来,现在得到许多应用的支持,包括Microsoft的Windows和Quick Basic、Ashton-Tate的Framework,以及操作系统OS/2。下拉式菜单激增的原因是新手非常容易使用它们。例如,在Quick C环境里,你可以上下移动亮条(highlight bar)进行选择,也可以按显亮字符进行选择。

在主菜单的File选择下面有一系列选择,这些选择通过下拉式菜单选用。参见图3。这12种允许的选择都与主选择File有关。例如,由这个下拉式菜单你能挑选打开一个已有的文件,建立一个新文件,保存你处理完的文件,和换一个名字保存你正在处理的文件。File下拉选择的大部分与文件有关。“Print”、“DOS Shell”和“Exit”实际上与文件无关,但是找不到比放在这里更合适的地方。

图4示出了Run下拉式菜单。头两个选择项允许向内存编译,或做些改变后重新编译。该编译选择使得在Quick C环境显示一个大的对话框,它接受各种编译选择标记。在对话框内能令Quick C生成一个可执行的文件。图5显示了编译对话框。

File和Run下拉式菜单是该系统最常用的。使用一个菜单打开和建立文件,另一个生成程序。其他菜单当你Quick C环境熟悉以后会用到。

1.3 使用Microsoft C

编译本书的程序时需要Microsoft C5.0或更高的版本,或者Quick C1.0或更高的版本。Microsoft编译程序使用CL命令把列出的源代码表编译成可执行的程序。它提供了比Quick C更大的灵活性和水平更高的最佳化,而且还保持与Quick C兼容。由于许多C的新手先有自己的Quick C,然后再提高,因此,这些程序按Quick C隐含的存储模式设计的,即中模式。你可以按你自己的需要改变存储模式,并以你选的存储模式重新编译它们。本书的所有程序不必改动用C5.0都能处理。

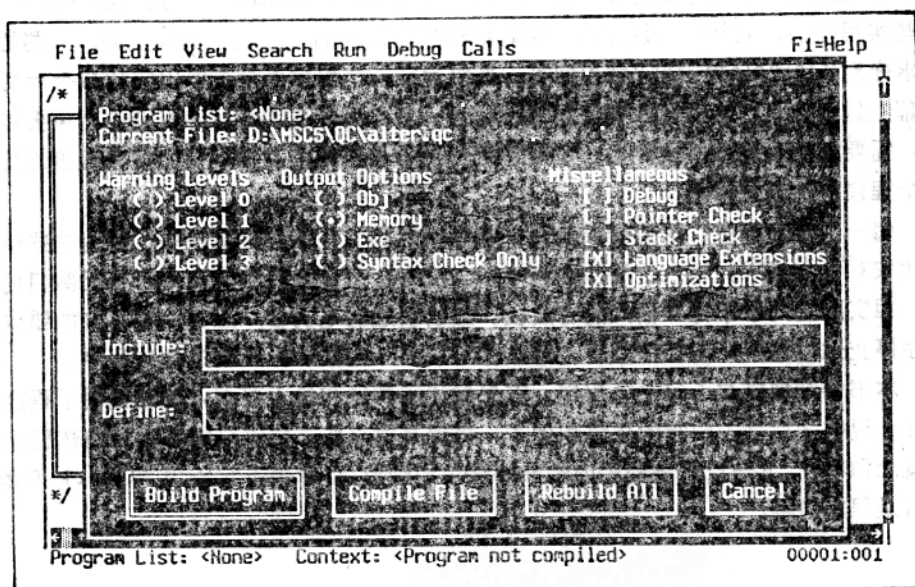


图5 编译对话框

CL命令不止能编译也能链接C源码。通过命令行开关，你能指定最佳化的级别，存储模式，甚至不让CL命令一步链接。（与其他程序链接建立目标文件时，令CL不链接是必要的。）编译本书里的程序的格式是：

CL /AM filename.c

接着斜杠的选择告诉编译程序把你的编译成的代码与中存储模式库链接。有许多选择可以使用，编译程序对它做解释。在DOS命令行敲CL/help会显示选择表。

1.4 与Microsoft C链接

上一段给出的CL命令足够以中存储模式编译本书的程序，并把它们与中存储模式库链接。然而，你感兴趣把这些实用程序结合到大的实用程序里或你自己研制的程序里时，还需要链接。LINK命令像这样：

LINK objname, exename, mapname, libraryname(s)

参数objname是编译程序生成的扩展名为.obj的文件。若链接不止目标一个文件，在第一个逗号前列出所有目标文件，它们之间用加号分隔。参数exename是要生成的可执行文件名。类似地，mapname是map文件的名字。map文件是由链接程序产生的文件，其装有关于可执行文件的组织的低级信息。链接用的库由最后一个参数给出。若有多个库，它们之间用加号隔开。

如有必要，可包括其他选择。这些细节超出了本书的范围。要得到完整的选择项表，请在命令行敲：

LINK/HELP

1.5 使用这些实用程序

本书的每一个程序都是命令行驱动的。这意味着在DOS提示下把大部分输入给予这些实用程序。通常敲入程序名，接着敲该程序需要的文件名或其他东西。一些程序也要求你对Yes或No提示敲Y或N回答。当你运行一个程序而没有给出它的命令行变元，屏幕上显示信息，告诉你如何使用它。信息一般一行：说明该程序在命令行需要多少变元，需要的变元是数、文件名、还是字符。由于信息是简要的，要想最大程度地使用每一个程序，请参见本书。

每一个程序都做广泛的错误检查。存储错误和文件错误检查最多，因为这些错误会产生灾难性的后果。输入合法性检查不多。所有这些程序都检查命令行变元数目的正确性，但实际上几乎不检查输入的合法性。敲错引起的毛病会产生文件找不到的错误，使程序停止运行而不产生任何损害。那么你可以加一些错误检查改进这些程序。

本书详细地描述了每一个程序。首先叙述了功能和输入要求，接着说明了程序的有用性。说明之后是关于它的用法和例子，再就是要点段。要点段解释和指出该程序的一般概念和特性。下一段是该程序内的具体程序的细节，或者关于它所用的Quick C函数。最后是程序清单。

第二章 目录实用程序

2.1 FILEATTR 列出目录里所有文件的属性

类型：外部的

用途：该实用程序显示与命令行给出的查找串匹配的文件的属性。它查找整个当前挑选的目录。

格式：C:\> fileattr filename.ext

其中filename.ext可以是具体的文件名，也可以是包括星号或问号的通配符文件名。也接受路径名。例如，下面的命令行是合法的：

C:\> fileattr c:\db4\filename.ext

注解

FILEATTR产生由硬盘或软盘任何目录里的文件的文件属性组成一张格式整洁的表。它速度快，与本书中的ALTER程序结合起来使用能改变文件属性。使用该程序能把当前的属性表输出到打印机或文本文件里，然后使用ALTER有选择地改变你要改的一些文件的属性。通过重定向该程序的输出到打印机或文件，能将这张表发送到打印机或文件。

例子

例子1:

C> fileattr *.*

显示当前目录里的所有文件以及它们的属性。

例子2:

C> fileattr *.exe

找到当前目录里扩展名为.exe的所有文件，并显示它们以及它们的属性。

例子3:

C> fileattr d:*.bak

找到驱动器D: 根目录下具有扩展名.bak的所有文件，并显示它们以及它们的属性。

例子4:

B> fileattr i:*.com> prn

找到网络驱动器I: 根目录下具有扩展名.com的所有文件，并且将通常发送到显示屏的信息重定向到打印机。

例子5:

C:\DOS> fileattr c:\db4\data*.qbf

FA-File Attribute, QC Edition, (C) Copr 1988, Robert Alonso
Path: C:\DB4\DATA

NAMES.DBF Archive
QUICK.DBF Archive

Number of files: 7

例子6:
C) fileattr
显示帮助信息。

编程要点

FILEATTR使用几个Quick C函数,对该程序最重要的一个是_dos_getfileattr。当给了它一个全文件名和扩展名,它返回一个错误,或者返回以一个无符号整数表示的该文件的属性。这个值必须由本程序分成几个有意义的部分,使得在屏幕上能打印出恰当的属性。使用AND (&) 算符检查返回值里的一些位的开和关状态完成这项工作。若看了源代码,你会看到变量FILEATTR和表示各种属性的定义进行的“与”操作。这些定义在头文件DOS.H里。

程序里也用到_dos_findfirst和_dos_findnext这对函数。命令行给出的查找串传送给这两个函数,并且也重复调用它们直到找出所有匹配文件为止。这两个函数返回的一部份数据是找到的文件的文件名。该程序使用这个信息在屏幕上显示文件名,并把这个名字传送给_dos_getfileattr函数。

子程序解释

函数_dos_getfileattr要求提供一个指向文件名的指针作为它的第一个变元。而第二个变元是一个无符号整数的地址,它用于存放返回的属性。若它成功的话,返回零值,并放文件的属性在第二个变元提供的地址里。否则返回错误代码,并且对Quick C变量ERRNO赋值。在DOS.H文件里有该函数的原型:

```
unsigned _dos_getfileattr(char *filename, unsigned *fattr);
```

程序

```

1  /*
2
3
4  Title           : FILEATTR.EXE           Version 1.0q
5
6  Version(s)      : 1.0q - March 30, 1988
7
8  Purpose         : Will fetch the attribute value of a file or group
9                   of files which match the DOS command line
10                  arguments. The output can be redirected and
11                  includes the current drive and path, the file name
12                  and extension of each file which matches and the
13                  attributes which are set.
14
15  I/O Requirements : The program will work on MS-DOS and PC-DOS
16                    computers using version 2.0 (of DOS) or greater.
17
18  Compiler Used   : Microsoft Quick C
19
20
21 */
22
23 #include <stdio.h>
24 #include <stdlib.h>
25 #include <dos.h>
26 #include <string.h>
27 #include <direct.h>
28
29
30 /*
31 only internal procedure used in this Program, since it
32 requires no input and returns nothing it is declared void for
33 both in going parameters and return values.
34 */
35
36
37 void filemode(void);
38
39
40 /*
41 Global variables used throughout this program.
42 */
43
44
45 char    searchstr [_MAX_PATH], pathname [_MAX_PATH],
46         fullname [_MAX_PATH], drive [_MAX_DRIVE],
47         subdir [_MAX_DIR], file [_MAX_FNAME],
48         ext [_MAX_EXT], holder [81];
49
50 int     done, count, problem;
51 unsigned fileattr;
52
53 struct find_t dta;
54
55
56 /*
57 The command line is parsed for a string which is later
58 used in a search for matching files. The FILEMODE

```

```

59 function performs all the work based on the SEARCHSTR.
60
61
62 CALLED FROM, (DOS)
63
64 CALLS, FILEMODE
65
66 */
67
68
69 main(int argc, char *argv[])
70 {
71
72 printf("\nFA-File Attribute, QC Edition, (C) Copr 1988, Ro-
bert Alonso\n");
73
74 if(argc > 1)
75 {
76 strcpy(searchstr, argv[1]);
77 _splitpath(searchstr, drive, subdir, file, ext);
78 sprintf(pathname, "%s%", drive, subdir);
79 }
80 else
81 {
82 printf("\nnusage: C:\> fileattr fullname.ext\n");
83 return(1);
84 }
85 filemode();
86 return(0); /* No problems encountered */
87 }
88
89
90 /*
91 This procedure searches for matches to the SEARCHSTR
92 and continues to find matching files until no more
93 are available. As each one is found, the file name
94 and attribute are printed on the screen. A count
95 of files is maintained and is printed at the end.
96
97
98 CALLED FROM: MAIN
99
100 CALLS: (NONE)
101
102 */
103
104
105 void filemode(void)
106 {
107
108
109 count = 0;
110 if(strlen(pathname) < 1)
111 getcwd(pathname, _MAX_PATH);
112 else
113 pathname[strlen(pathname) - 1] = 0; /* get rid of extra s\
ash */
114

```

```

115strupr (pathname);
116printf ("\\npath: %s\\n\\n", pathname); )
117strcat (pathname, "\\"); /* put slash back now */
118
119done = _dos_findfirst (searchstr, 47, &dta); /* all including hidden */
120
121while (! done)
122{
123    count++;
124    strcpy (fullname, pathname);
125    strcat (fullname, dta.name);
126
127    problem = _dos_getfileattr (fullname, &fileattr);
128    if (! problem)
129    {
130       strupr (dta.name);
131
132        printf (" %-12s", dta.name);
133        if (fileattr & _A_NORMAL)
134            printf ("Normal");
135        if (fileattr & _A_RDONLY)
136            printf ("Read Only");
137        if (fileattr & _A_HIDDEN)
138            printf ("Hidden");
139        if (fileattr & _A_SYSTEM)
140            printf ("System");
141        if (fileattr & _A_VOLUME)
142            printf ("Volume");
143        if (fileattr & _A_SUBDIR)
144            printf ("Subdirectory");
145        if (fileattr & _A_ARCH)
146            printf ("Archive");
147
148        printf ("\\n");
149    }
150    else
151        printf ("a %-12s - ERROR! \\n", dta.name);
152
153    done = _dos_findnext (&dta);
154}
155
156if (count == 0)
157    printf ("No files match the search string.\\n");
158else
159    printf ("\\nNumber of files: %3d\\n", count);
160}

```

源程序的译文

8—13行

用途：取与DOS命令行变元匹配的一个或一组文件的属性。输出可以重定向。并且输出中包括当前驱动器 and 路径、每个匹配文件的名称和扩展名，以及对文件指定的属性。

31—33行

该程序用的惟一内部过程。由于它不需要输入和返回，因此它的输入和返回值都说

明为VOID。

57—59行

对命令行进行词法分析找出以后查找匹配文件要使用的串。FILEMODE函数基于SEARCHSTR进行所有的工作。

91—95行

该过程查找与SEARCHSTR匹配的文件，直到找出所有的匹配文件为止。每当找到一个，文件的名字和属性定义显示在屏幕上。管理一个文件个数计数器，在该过程最后打印出它的值。

2.2 FILESIZE 列出目录里所有文件和它们的规模

类型：外部的

用途：该实用程序显示出与查找串匹配的一组文件中每一个的规模。也打印出所有匹配文件的总数。此外，该程序也给出你指定驱动器上磁盘空间总量和可使用的空间量。

格式：C:\> filesize filename.ext

其中filename.ext可以是具体的文件名，也可以是包括星号或问号的通配符文件名。也接受路径名。例如，下面的命令是合法的：

C:\> filesize c:\qa\filename.ext

注解

FILESIZE使你能快速地确定一组文件占有多大空间。然后，使用这个信息判定是否这些文件放在一张磁盘上。该程序要求给它用来查找匹配文件的具体文件的名字，或者具有通配符的文件名。在每种情况，程序都打印出找到的文件的名字和规模。

该程序一结束处理就打印出关于文件和指定驱动器的统计数据。你会得到文件的总数以及它们所占的字节数。在另一行上，程序显示该驱动器的空间总量和还没有用的空间。打印出的统计类似于：

```
Total of 27 files is:      452223 bytes
Drive usage:
Total space on drive:      33462272 bytes
Space not yet used:        264192 bytes
```

例子

例子1:

C> filesize *.*

显示当前目录里的所有文件和磁盘消耗的统计数据。若没有指定的驱动器，该程序

认为你要的是隐含驱动器上的总空间和未用的空间。在该例中隐含驱动器是驱动器C。

例子2:

```
C> filesize *.wks
```

找到当前目录里扩展名为.wks的所有文件,并显示它们的规模统计数据。

例子3:

```
C> filesize d:\*.bak
```

找到驱动器D:根目录下具有扩展名.bak的所有文件,并显示规模和D:驱动器上的空间统计数据。

例子4:

```
B> filesize h:\*.doc > prn
```

找到网络驱动器H:根目录下具有扩展名.doc的所有文件,并且将通常发送到显示屏的信息重定向到打印机。

例子5:

```
C:\DOS> filesize c:\alpha3\sample\*.dbf
```

FS-File Size, QC Edition, (C) Copr 1987, Robert Alonso

NAMES.DBF	20911
-----------	-------

QUICK.DBF	7299
-----------	------

Total of 2 files:	28210 bytes
-------------------	-------------

Drive usage:

Total space on drive:	33462272 bytes
-----------------------	----------------

Space not yet used:	264172 bytes
---------------------	--------------

例子6:

```
C> filesize
```

显示帮助信息。

编程要点

FILESIZE使用几个功能强的Quick C函数。其中的一对是dos_findfirst和dos_findnext函数,另一个是dos_getdiskfree函数。前两个已经描述过,但还有几句话要说。命令行给出的查找串传送给函数dos_findfirst和dos_findnext,并且这两个函数被重复调用直到找出所有匹配文件为止。这两个函数返回的一部分信息是找到的文件的规模。该程序使用这个信息在屏幕上打印文件的规模,并把它加到管理运行总量的计数器上。

该程序后来使用dos_getdiskfree取得指定磁盘驱动器的容量和该磁盘还未用空间的总量。这两个结果在程序结束处打印出来。

该程序的特性之一是找到隐藏的文件。你修改`_dos_findfirst`函数调用的数47 可以改变这一点。此数表示该函数要查找的文件的属性。若这个数改成41, 程序会跳过隐藏的文件和系统文件。关于为什么这样工作的细节, 请读VWIPE程序这一节。

子程序解释

`_dos_getdiskfree`函数要求你传给它代表隐含驱动器的0, 或者代表驱动器字母的一个具体数。A: 驱动器是1; B: 驱动器是2; C: 驱动器是3。一般地, 驱动器字母 越大, 相应的数也越大。例如, D: 驱动器是4。为了使该程序的能力更强, 采取了几个步骤得到用户在命令行上指定的惟一驱动器上的未用空间。

`_splitpath`函数用来得到驱动器字母。如果驱动器字母存在, 那么从它里减去值94。原因是A的ASCII值是65。如果你将65减去64, 那么结果是1。该值 恰好是`_dos_getdiskfree`函数需要的。如果没有指定驱动器, 对于变量`DRIVENO`什么也不做。由于该变量自动地初始为0, `_dos_getdiskfree`函数认为0代表隐含驱动器, 因而不会出现任何问题。

词汇表中的词

网络 重定向 ASCII

程序

```
1  /*
2
3
4  title           : FILESIZE.EXE      Version 1.0q
5
6  Version(s)      : 1.00 - September 29, 1987
7                  1.0q - February 26, 1988
8
9  Purpose         : Goes through an entire directory and displays all
10                  the files in it. Next to the file listing, the size
11                  is also listed. At the end of the display, the
12                  utility prints totals of all the files found.
13
14  I/O Requirements : The program will work on MS-DOS and PC-DOS
15                  computers using version 2.0 (of DOS) or greater.
16
17  Compiler Used   : Microsoft Quick C
18
19 */
20
21
22 #include <stdio.h>
23 #include <stdlib.h>
24 #include <dos.h>
25 #include <string.h>
26
27
28 /*
29 The one procedure which is internal to this program is
30 called by the same name as the program.
31 */
32
33
34 void filesize (void);
35
```