

5.0 版

TURBO C TOOLS

使用手册

希望公司



HOPE

中国科学院
希望高级电脑技术公司

中国科学院希望高级电脑技术公司

一九九〇年三月

前 言

Turbo C 工具库为一般 Turbo C 程序提供一系列函数和工具。Turbo C 用户可以使用工具库中的函数实现屏幕、键盘、窗口、菜单、热键、图象、存贮空间、伪脱打印机以及文件和目录的多功能控制和管理；并能编写软硬件中断服务程序，调用 DOS 服务，管理常驻内存程序。跟踪错误，实现点操作、字符串转换、直接视屏访问等功能。

Turbo C TOOLS 完全适用于 Turbo C 环境，能在所有运行 Turbo C 的机器上运行，并支持各种输入输出设备。而且由于 far 关键字和 near 关键字的开发，从而支持六种存贮模式。Turbo C ~~TOOLS~~ 易于理解和使用，并易于调整以适于用户的不同需要。

本书详述 Turbo C TOOLS 5.00 的各种功能，~~并对其中的所有函数~~都作了详尽的描述，为用户使用 Turbo C ~~TOOLS~~ 提供指导和帮助。

本书由译者在长期从事 Turbo C 程序设计的基础上，经认真翻译、整理而成，但由于译者水平有限，加上时间仓促，错误缺点在所难免，欢迎读者指正。

在本书翻译过程中，得到了叶欣同志的大力帮助和支持，译者在此表示衷心的感谢。

吕良双、汪亚文、罗红斌、凌学焦、占卫兵、谢康、汤道志等同志为本书的出版做了大量工作。

译 者

1990 年 1 月 20 日

目 录

简介	1
概述	1
Turbo C TOOLS 函数的分类	1
手册的组织	2
软件需求	3
硬件需求	3
Turbo C TOOLS 的使用	4
磁盘里的内容	4
安装	4
创建第一个程序	5
编程风格、编译警告和头文件	6
修改 Turbo C TOOLS	6
源代码约定	7
打印源代码	7
字符串函数 (ST)	9
字符串函数分类	9
屏幕函数 (SC 和 VI)	10
IBM 视频设备的功能	10
屏幕处理函数的分类	12
高级问题	14
如何在显示设备间切换	14
怎样管理多显示页	14
保存和恢复视频状态	15
直接屏幕访问的限制	15
图形函数 (GR)	17
图形函数的功能	18
窗口管理 (WN)	19
窗口功能简介	19
一个简单的例子	19
窗口函数的分类	21
使用更高级的窗口特征	23
不可见窗口	23
滞后和更新窗口	23

窗口的光标	23
堆积窗口	23
不可删去窗口	24
控制内存分配	24
错误处理	24
多显示页和设备	25
菜单函数 (MN)	26
菜单功能概述	26
菜单函数的分类	28
更高级的菜单特征	29
菜单的快速显示	29
菜单项的颜色	29
击键的功能和动作	29
在菜单中使用窗口函数	31
错误处理	32
键盘函数 (KB)	33
键盘函数的功能	34
文件处理函数 (FL)	35
文件处理函数的分类	35
打印函数 (PR)	36
打印函数的分类	36
存贮管理 (MM)	37
DOS 存贮概念	37
DOS 存贮控制块结构	37
存贮管理函数	38
中断服务 (IS)	39
介绍	39
背景和术语	39
ISR 的安装	39
中断服务函数	40
创建和使用 ISR	41
调用顺序	41
安装	42
删除	42

设计 ISR	42
促进可重入	43
中断过滤	43
硬件中断优先级	44
ISR 的功能限制	45
前台和后台程序以及它们的特权	45
后台程序中的浮点运算	45
避免 DOS 重入问题	46
中断 8 用于后台调度	46
ISR 特殊出口	47
插入码 (IV)	48
介绍	48
插入码机制	48
插入机制的实用函数	49
创建插入函数	49
调度	49
调用顺序	50
安装	51
重调度	51
删除	51
实用函数和宏 (UT)	53
实用函数的分类	53
函数参考说明	56
FLDOLOCK	56
FLFLUSH	57
FLGETDTA	58
FLLOCK	59
FLNORM	60
FLPROMT	63
FLPUTDTA	64
FLREMOVOL	64
FLRETVOL	65
FLSETVOL	66
GRAMOVE	67
GRLINE	67
GRPTREAD	68
GRPTWRIT	69

GRRMOVE	70
ISCALL	71
ISCURPRC	72
ISGETVEC	73
ISINSTAL	74
ISPREP	76
ISPUTVEC	77
ISREMOVE	78
ISRESEXT	79
ISSENSE	79
IVCTRL	80
IVDISABL	81
IVINSTAL	82
IVSENSE	84
IVVECS	85
KBFLUSH	86
KBGETKEY	87
KBPLACE	88
KBQUERY	89
KBQUEUE	90
KBREADY	91
KBSCANOF	92
KBSET	93
KBSTATUS	94
KBSTUFF	95
MMCTRL	96
MMSIZE	97
MNCREATE	98
MNDSPLAY	99
MNDSTROY	100
MNITEM	101
MNITKEY	102
MNKEY	103
MNLITEM	104
MNLITKEY	105
MNLREAD	106
MNREAD	108
PRCANCEL	109
PRCHAR	110
PRERROR	111

PRGETQ	111
PRINT	113
PRINSTLD	113
PRSPPOOL	113
PRSTATUS	114
SCAPAGE	115
SCATTRIB	116
SCBLINK	117
SCBORDER	117
SCBOX	118
SCCHGDEV	120
SCCLRMSG	121
SCCURSET	121
SCCURST	122
SCEQUIPBR	122
SCMODE	123
SCMODE4	124
SCNEWDEV	125
SCPAGE	125
SCPAGES	126
SCPALI	127
SCPALETT	127
SCRCLR	129
SCPGCUR	129
SCREAD	130
SCROWS	130
SCTTYWIN	131
SCTTYWRT	132
SCWRAP	133
SCWRITE	135
STPCVT	135
STPEXPAN	137
STPJUST	138
STPTABFY	139
STPXLATE	139
STSCHIND	140
UTCRT	141
UTCTLBRK	142
UTDOSRDY	143
UTGETCLK	143

UTINTFLAG	144
UTMODEL	145
UTMOVEMEM	146
UTNORM	147
UTNULCHK	148
UTOFF	149
UTPEEKB UTPEEKN UTPEEKW	150
UTPLONG	152
UTPOKE UTPOKEN UTPOKEW	152
UTSEG	153
UTSLEEP	153
UTSPKR	154
UTTOFAR UTTOFARU	154
VIHORIZ	155
VIPTR	156
VIRDRECT	156
VISROLL	157
VIWRRECT	158
WNATRBLK	158
WNATRSTR	159
WNATTR	160
WNCREATE	161
WNCURMOV	161
WNCURPOS	162
WNCURSOR	162
WNDSPRAY	163
WNDSTRDY	166
WNERORR	166
WNGETOPT	167
WNHORIZ	168
WNQUERY	169
WNRDBUF	170
WNRWNREDRAW	171
WNREMOVE	172
WNSCRBLK	173
WNSCROLL	174
WNSELECT	175
WNSETBUF	176
WNSETOPT	176
WNUUPDATE	177

WNWRAP·····	178
WNWRBUF·····	179
WNWRSTR·····	181
WNWRSTRN·····	181
WNWRTTY·····	183
附录 A 错误类型、原因及处理·····	184
附录 B 头文件·····	190
附录 C 显示模式·····	192
附录 D 屏幕/窗口比较·····	193
附录 E 窗口和菜单错误码·····	195
附录 F 全局变量·····	196
附录 G 内部函数·····	197
附录 H IBM 及 DOS 依赖性·····	199
附录 I 开发注意事项·····	200
附录 J 样例程序·····	201

简介

概述

Turbo C TOOLS 是一函数包，其中的函数能用于 **IBM** 个人机家族的 **Turbo C** 程序中。它们提供：

- 屏幕和键盘的快速、多功能控制；
- 创建、显示和使用多窗口的窗口管理；
- 使用移动高亮条的用户界面的菜单；
- 完全用 **C** 语言编写中断服务程序的能力。这些程序处理硬件和软件中断，并调用 **DOS** 服务；
- 插入码：按一热键，或在给定的时间间隔里调度 **C** 函数执行的能力；
- 创建和删除驻留在内存中的程序（也叫 **TSR** 应用）的能力；
- 常驻伪脱打印机制的界面；
- 具有强功能翻译和转换的常规字符串函数。

Turbo C TOOLS 经过精心设计、多年鉴定，证明是适用于 **Turbo C** 环境的。**far** 键和 **near** 键被开发用于创建在所有六种存贮模式下都能有效工作的实用函数和宏。**Turbo C** 函数在原型开发中得到了大量的使用，使得数据类型错误在开发周期中尽可能早地检测出来。

Blaise 计算机库产品的连续精化已产生了一组工具，其可靠性、功能性和易使用性都不太好。而 **Turbo C TOOLS** 却具有如下的高级开发特性：

- 指针操作宏。这些宏能访问存贮区中的任何位置，计算指针或地址的段地址和偏移量，或者创建指向任何数据类型的远指针。这些宏还能使用户建立在任何存贮模式下都有效的程序，即用户不用关心与模式有关的指针大小。
- **UTNULCHK** 和 **UTCHKNIL** 对空指针赋值的智能检测。这些简单但有效的工具使用户能跟踪最普遍、最难以捕捉的错误：没有初始化的指针的使用。
- 函数 **FLNORM** 能将任何文件名或路径名变换为标准形式，以便与其它文件名相比较。

Turbo C TOOLS 经过精心设计，支持许多易使用的重要的功能。整个源代码组装在一起，使用户能调整此程序包适应自己的需要。另一方面，用户使用 **Turbo C TOOLS** 时不必考察其源代码，除非有意这么做。若用户需要调整其中的功能以适应自己的特殊需要，将会看到 **Turbo C TOOLS** 的源代码是易于理解、修改和安装的。

Turbo C 是软件开发的优秀编译器。写得好的 **C** 程序是可靠的、有效的、可读性好的、易于维护的。并且 **Turbo C** 代码量少，运行速度快。然而，许多功能需要利用 **IBM PC** 环境中的高级硬件和软件特性才能实现，标准 **Turbo C** 库不支持这些高级功能。而 **Turbo C TOOLS** 支持许多附加的高级功能。

Turbo C TOOLS 函数的分类

Turbo C TOOLS 函数根据各工具解决的问题或相关问题集分类。共有 13 类：

前缀	类别
ST	字符串函数
SC	屏幕处理
VI	直接视频访问
GR	图形
WN	窗口管理
MN	菜单
KB	键盘控制
FL	文件和目录管理
PR	打印机和PRINT.COM伪机打印机界面
MM	存贮管理
IS	中断服务支持
IV	插入码
UT	实用函数和宏

表中的前缀贯穿全文档和源代码，用作标识函数类别。每个函数的名字以分类前缀打头。例如，屏幕处理类（前缀‘SC’）包括在屏幕上画框的函数 `scbox()`。象所有 Turbo C TOOLS 函数一样，调用它时使用它的小写字母名字，它的手册中和其他文档中的名字为大写 SCBOX。

虽然，每类函数解决一个特定领域的问题，但 Turbo C TOOLS 中有些函数依赖于别的函数。我们力图使各类函数自包含，以便你不用研究整个产品，就可以学习使用其中的一类函数。然而，有些领域的问题不可避免的需要别的领域的知识。例如，为了充分发掘窗口管理函数的功能，有必要了解屏幕处理函数的功能和限制。

手册的组织

介绍后的一章是“使用 Turbo C TOOLS”。阐述安装、使用和修改 Turbo C TOOLS 必需的步骤。有关 Turbo C TOOLS 的特定版本的指导信息列在盘上的 `readme.doc` 文件中。此文件附属于本手册，请参阅它。

接下去的各章分别描述 Turbo C TOOLS 的各类函数。（SC 和 VI 类合成一章。其它各类独自成章）。阅读这些章节，你可以了解 Turbo C TOOLS 提供的服务，并学会选择最满足需求的函数。

手册的最大部分是函数的参考部分。Turbo C TOOLS 的几乎每个函数都在其相应参考“页”中作了详尽介绍。参考页以函数名的字母顺序排序。

杂项参考信息列在附录中。你不必知道这些信息，但要知道怎样查找。特别是要了解附录 A 中的错误诊断指导。下面是各附录的主要内容：

附录 A 是错误表和提示信息。若你的程序失败或遇到不理解之处可以查阅它。

附录 B 描述头文件（.h 和 .mac），包括 `compile.mac` 的使用。

附录 C 包含 IBM 视频模式表。

附录 D 标明屏幕处理（SC）和窗口管理（WN）类中的并行函数。为根据需求查找适

当函数提供指导。

附录 E 列出窗口 (WN) 和菜单 (MN) 函数返回的错误代码的值和意义。

附录 F 列出 Turbo C TOOLS 函数定义的所有全局变量。

附录 G 描述仅内部使用的 Turbo C TOOLS 函数。

附录 H 指出需要 IBM 软件或 DOS 3.00 以上版支持的函数。

附录 I 描述用于开发 Turbo C TOOLS 的环境。

附录 J 是一些简单程序的文档。

软件需求

我们假定你熟悉 PC-DOS 上运行的 Turbo C, 以及由 Kernighan 和 Ritchie 描述的 C 语言。最初版本的 Turbo C TOOLS 支持 Turbo C 1.00。关于以后版本的信息请参阅盘上的 readme.doc 文件。

有些 Turbo C TOOLS 函数是用汇编语言写的。一般不需汇编器, 除非修改这些特殊函数。若一定要这样做, 建议用 Microsoft 的 3.00 以上版本的宏汇编包。我们编译 Turbo C TOOLS 汇编语言时也使用它。

为了推广和维护目标模块库 (.lib), 需要有与 Microsoft 库管理软件 LIB 兼容的库管理程序。这也由 Microsoft 宏汇编包提供。

为了推广可执行 .exe 文件, 我们用到了 Turbo C 编译中的 TLINK 程序。

还需要 DOS 2.00 或更高版本, 尽管有些函数只适用于 3.00 以上版本, 但大部分函数在 DOS 的所有版本中的行为相同。需要 DOS 3.00 以上版的函数列在附录 H 中。

硬件需求

许多 Turbo C TOOLS 程序能在运行 Microsoft 操作系统 (MS-DOS) 2.00 以上版本的机器上正确执行。(IBM PC-DOS 是 Microsoft MS-DOS 的改写版。) 事实上, ST、FL、MM 或 IS 类中的程序都不依赖于任何特定的 IBM 硬件特性。

附录 H 列出了需要 IBM 兼容硬件支持的所有 Turbo C 程序。屏幕 (SC 和 VI)、图形 (GR)、窗口 (WN) 和菜单 (MN) 函数需要 IBM PC 家族的可兼容 BIOS 界面和相同的视频适配器地址。一些键盘程序用到 BIOS 调用, 另一些需要支持键盘缓冲区的有关信息。插入码机制 (IV) 需要 IBM PC 及兼容机的特殊硬件中断结构。

使用 Turbo C TOOLS 要两个盘驱动器 (最小是两个 360K 的盘驱动器)。用硬盘更好。Turbo C TOOLS 的存贮需求与 Turbo C 的不同, 但这种支持越多越好。在开发周期中最好有足够空间安装 RAM 盘或虚拟盘。

Turbo C TOOLS 的使用

本章阐述在应用环境中安装和使用 Turbo C TOOLS 函数的必要步骤。这里的描述通用于所有的存贮模式。磁盘上的 `readme.doc` 文件包括 Turbo C TOOLS 特定版本的信息，可以看成是本章的补充，你要好好阅读它。

磁盘里的内容

磁盘中提供由 Turbo C 使用 Turbo C TOOLS 必需的源代码和库。盘上的 `readme.doc` 文件列出了提供的所有文件和它们的使用说明。包括指明哪些函数是用 C 语言写的，哪些是用汇编语言写的，哪些是头文件中的宏定义。

盘上的文件遵从一致的命名规则。它们使用如下的文件名扩展字：

文件名扩展字	描述
<code>.c</code>	C 函数源文件
<code>.h</code>	头文件：数据定义
<code>.asm</code>	汇编语言源文件
<code>.mac</code>	汇编语言头文件
<code>.obj</code>	目标代码文件
<code>.bat</code>	DOS 批处理文件
<code>.doc</code>	文档文件
<code>.lib</code>	Turbo C TOOLS 函数库
<code>.fil</code>	Turbo C TOOLS 源文件名表
<code>.exe</code>	可执行程序

当文件有多种版本以支持不同的编译版本和存贮模式时，使用下面的规则：给文件加一个 3 个字符组成的后缀。后缀的前两个字符表示编译器及其版本：

T1	Turbo C 1.00 版
其它	(见盘上的 <code>readme.doc</code> 文件)

后缀的最后一个字符表示存贮模式：

T	微模式	(64K 连在一起的代码和数据)
S	小模式	(64K 代码, 64K 数据)
M	中模式	(64K 代码, 1M 数据)
C	紧缩模式	(1M 代码, 64K 数据)
L	大模式	(1M 代码, 1M 数据)
H	巨模式	(1M 代码, 1M 数据 --- 数据项可以超过 64K)

其它 (参见盘上的 `readme.doc` 文件)

例如，`tct_t1s.lib` 是 Turbo C 1.00 版、S 模式的 Turbo C TOOLS 库，而 `comp_t1m.mac` 是 Turbo C 1.00 版、M 模式的 `compiler.mac` 的版本。(关于 `compile.mac` 的更多的信息参见附录 B)。

安装

首先根据手册中的说明安装 Turbo C 的编译器。必须遵从下列步骤 (熟悉后，你可以

改变目录结构，使之符合你自己的口味）。

- 拷贝 `tc.exe`（集成环境）和/或 `tcc.exe`（命令行版）到软盘或硬盘上。为了方便，可以把它们放到一个由 `PATH` 命令指定的目录中（一般是将所有的编译器和其它可执行 `.exe`、`.com` 程序放到 `\bin` 目录中）。
- 创建一工作目录或工作盘，包含所有的 C 和汇编源文件。如果用的是硬盘，创建 `\turbo` 子目录，如果用的是软盘，则建立一工作盘用于此目的。创建一目录包含所有的头文件（`.h` 文件）。如果你用的是软盘，则叫此目录为 `\include`。如果你用的是硬盘，则创建目录 `\turbo\include`。
- 将所有的标准 Turbo C 头文件和 Turbo C TOOLS 头文件拷贝到头文件目录中。
- 将头文件目录的名字告诉给编译器。在集成环境中，用 `Option/Environment` 菜单做到这一点。对命令行版，在你的配置文件 `tcconfig.cfg` 中加上如下行（可用你自己的目录名替代）：

`-Ic: \turbo\include`

- 创建一目录包括所有的库（`.lib`）文件，如果用的是软盘，叫它 `\lib`，如果用的是硬盘，则创建目录 `\turbo\lib`。
- 拷贝标准的 Turbo C 库和初启目标模块（`c07.obj`）到库文件目录中。然后将合适的 Turbo C TOOLS 库（`tct_t1?.lib`）拷贝到此目录。
- 将你的库文件目录名告诉给编译器。在集成环境中，用 `Options/Environment` 菜单做到这一点，对于命令行版本，则在你的 `tcconfig.cfg` 文件中加上如下行（可用你自己的目录名替代）

`-Lc: \turbo\lib`

- 置所有的编译器警告消息为使能（Enable）。这将给许多种细微错误提出早期警告。在集成环境中，使用 `Option/Compiler/Errors` 菜单。对命令行版本，在你的 `tcconfig.cfg` 文件中加上下列行：

`-w`

- 如果你用的是集成环境，则通过 `Option` 菜单在 `tcconfig.tc` 文件中保存你的选择。

创建第一个程序

Turbo C TOOLS 通过精心设计，不需要特殊的编译和联结任选项。你需要关心的只是：

- 包含适当的头文件；
- 通过适当的编码风格消除所有的编译警告；
- 与适当的 Turbo C TOOLS 库（`tct_t1?.lib`）联结。在用 Turbo C TOOLS 创建第一个程序之前，你应该首先创建和执行一个或多个没有使用 Turbo C TOOLS 的简单程序，以此来练习编译。试一下传统的测试程序：

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hello, World!\n");
```

```
    return 0;          /* Return code 0 indicates success. */
```

}

用 Turbo C TOOLS 创建程序时, 唯一附加的步骤是指明适当的 Turbo C TOOLS 库 (tct_t1?.lib)。对 Turbo C 的命令行版本 (tcc.exe) 来说, 仅仅指明库名即可, 如下例:

```
tcc wnexampl tct_t1s.lib
```

在集成环境中, 你必须用 Project menu 指明一项目文件, 以保证 Turbo C TOOLS 库与你的程序联结。你可以试试下面的简单项目文件:

```
wnexampl.c
```

```
tct_t1s.lib
```

作为开始的经历, 从 Turbo C TOOLS 盘中将 wnexampl.c 拷到你的工作目录或盘中。编译、联结、运行之。它能测试你是否安装好了 Turbo C 和 Turbo C TOOLS。如果此过程进行得很顺利, 你可以继续创建你自己的程序了。

编程风格、编译警告和头文件

现代 C 编译如 Turbo C 提供强数据类型的任选项。它们能以象 UNIX 的 lint 机制那样的方式辨别不当的数据类型转换错误。另外, 还能接受函数参数类型的说明, 并用此信息鉴别函数的数量和类型的不匹配错误。

强数据类型为检查和避免程序错误提供强有力的帮助。因此, Turbo C TOOLS 大量使用头文件来定义数据类型, 并声明函数的参数和返回值的数据类型。

建议你也这样做。置所有的编译警告消息为使能。包含所有有关的头文件, 这些头文件为你调用的 Turbo C 和 Turbo C TOOLS 库中函数声明参数。用自己的头文件声明自己定义的函数的参数。如果编译警告良性的类型冲突, 则仔细修改代码以避免或抑制警告。

如果你依靠头文件定义模块间共享的所有定义, 并且避免代码产生特殊编译错误, 那么你将从某些难以捉摸的错误的早期诊断中获益。(强类型甚至能诊断出还没有对你构成威胁但将来会对你有害的错误!)。

修改 Turbo C TOOLS

重新编译或重新汇编 Turbo C TOOLS 可以使它们适用于不同的编译版本和存贮模式。这在 readme.doc 文件中有详尽说明。(例如: 在头文件中被当作宏处理的函数)。一般的过程是:

为了修改一 Turbo C TOOLS 函数, 只需重新编译或重新汇编相应的源代码。所有源文件的名字都是函数名加上 .c 或 .asm 的扩展部分。readme.doc 文件指出哪些函数是用 C 语言写的, 哪些是用汇编语言写的, 哪些被当作宏来处理。

重新编译和汇编后, 再在编译和连接你的测试程序时显式指定目标文件。例如, 如果你修改了 STPCVT 并且创建了 stpcvt.obj, 用下面的命令将测试程序与 Turbo C 的 S 模式连接起来:

```
tcc test stpcvt.obj tct_t1s.lib
```

这样显式地命名 stpcvt.obj, 使连接器使用新版本代替库 (tct_t1s.lib) 中原来的版本。为了在集成环境中达到同样的目的, 在你的项目文件中加入下列行:

```
stpcvt.obj
```

若你对这种修改满意，可以用它取代库中的目标模块，这一工作需要一库管理程序完成，象 Microsoft 的 LIB 工具，它是由 Microsoft 宏汇编包提供的。

建议你不要改变任何 Turbo C TOOLS 函数的调用顺序。如果真的这样做，你会混淆可能调用这些函数的其它 Turbo C TOOLS 函数。相反，你可以自己命名定义新函数。

源代码约定

为了提高代码的易读性和可理解性，Turbo C TOOLS 中用到了一些约定。

每个函数名以一个两字母的前缀打头，它指明此函数的类属。（参见手册的介绍部分中的函数分类和前缀表）

每个函数的源文件的开头是一个文档头。调用序列、参数定义和说明信息都在源文件的开始用注释表达出来。

源代码遵从类似于 Kernighan 和 Ritchie 的著作里推荐的风格规范。所有的标识符都有声明，即使是整数也如此。指针变量以字母 P 打头；指针的指针以 PP 打头。缩进的风格也是标准的。编码风格虽是个人的事情，但是我们力图采用相对的标准，以提供一个可以让 Turbo C TOOLS 用户模仿的模式。

打印源代码

Turbo C TOOLS 经过了很好的设计和文档说明，容易理解，你不必参阅源代码。然而，你也许希望能很容易地打印代码。

为了打印源代码，我们提供了一个叫 CONCAT 的程序，它产生格式化的打印文件。你会发现它是对你的软件库的有益补充。格式化打印文件的每页的顶部是文件名、最后修改日期和时间以及页号。每页的第二行显示 CONCAT 格式化此文件的时间。每个输出行以一个换页字符（'\14'）结束。

为了执行 CONCAT，键入命令：

```
Concat /o ofile /p psize /m marg /n pnun /d debug
```

其参数的意义如下：

/o Ofile 是由 ofile 产生的格式化打印文件的名称。若此文件已存在，则被冲掉，若此文件不存在，则创建之。若没有给出输出文件名，则在当前目录下打开 concat .prt 文件。

/p Psize 是每页打印的行数。每页的头部需要 4 行（一个空白行，2 个头信息行，和另一个空白行）；因此最小的可接受值是 5。缺省值是 64，对标准 66 行来说，允许在每页的顶部有两个空白行。

/m Marg 是每个打印行前的空格数。缺省值是 0。最大值是 10。

/n Pnum 是开始页号。文件的各页顺序编号。每个文件的开始页号的缺省值是 1，如果 /n 没有指定值，则页编号顺序从 1 开始，但是再也不为每个文件重新设置开始页号。

/d Debug 是纠错信息送到其中的某文件的名称。如果此开关没有指定一文件名，则自动创建文件 concat.dbg。若没有指定此开关，则不产生纠错文件。

所有的参数都是任选项。如果用到开关，则参数可以任何次序出现。如果没有用到开关，则参数以上述顺序解释。开关字符的斜杠 (/) 可以用减号 (-) 代替。

例如，下面两命令中的每个都能产生输出文件 `foo.prt`，其格式是每页 64 个输出行，左边的空边为 10 个字符宽，首页号是 1：

```
concat foo.prt 64 10 1
```

```
concat foo.prt /n /m 10
```

执行 `CONCAT` 时，屏幕上显示一欢迎信息，然后出现提示符 `>>`。这时你可以键入一个文件的全路径名。此步骤可以重复任意次。一空反应（仅敲 `ENTER` 键）结束 `CONCAT`。

输出文件构造出来后，可以用 `DOS PRINT` 命令打印之，或拷贝到 `prn`（打印机）设备上。

`CONCAT` 展开 `tab` 字符成 8 个字符宽的空格，此约定在 `DOS PRINT` 和 `TYPE` 命令中也用到。

找不到输入文件时显示一警告信息，但是整个过程仍继续下去。`CONCAT` 在输出文件打不开或有错误写到输出文件中时才结束。