

第一章 字处理软件的使用技巧

第一节 Wordstar

WS的启动方式

原版 Wordstar 4.0 版提供了以下几种启动方式:

1. 键入“WS”，进入主菜单；
2. 键入“WS <filename>”，直接打开由 filename 指定的文件，进入文本编辑状态；
3. 键入“WS <filename> D”，同上；
4. 键入“WS <filename> N”，直接打开由 filename 指定的文件，进入标准文本编辑状态；
5. 键入“WS <filename> P”，启动 WS，自动打印 filename 指定的文件；
6. 键入“WS <filename> PX”，启动 WS，自动打印 filename 指定的文件，打印结束后退回 DOS 状态。以上两种方法均按缺省参数打印。

中文 WS 在启动时，若命令行给出了文件名，则直接进入文本编辑状态，也可修改 WS.COM，使其进入标准文本编辑状态。

```
C>DEBUG WS.COM
-E0392 FF
-W
-Q
```

如果要修改一个盘上的某一文件，而要求将结果存入另一盘中，则在启动 WS 时输入：

```
C>WS A:<filename> B:
```

则 A 盘上的由 filename 指定的文件将被换名为.BAK 文件，修改得到的文件用原名存入到了 B 盘中。

字符的替换

WS 有很强的字符查找和替换功能。巧妙地使用这种功能，可以加快编辑速度。

在编辑中文文章时常常要用到顿号、句号等键盘上没有的符号，而许多汉字操作系统只对其提供了区位输入方法，因此在输入这些符号时，只得切换输入状态，转到区位方式下，频繁地切换输入状态，势必降低输入速度。解决的方法是先将这些符号用键盘上很少用到的键的组合来代替，如用“####”代替句号。在输入完成后，键入<F5>键，屏幕将提示“找？”，输入“####”，再按回车，又提示“换成？”，用区位方式输入“。”，键入回车，屏幕最后提示“选择？”，键入 GN，表示自动查找并替换，键入回车后，文章中的“####”均被替换成中文方式的句号。

为了保证替换的准确性和彻底性，替换前应将光标移到文章的首行。若移到末行，就必须在以上的最后一个提示中，增加输入“B”，告诉 WS 进行逆向查找和替换。同时要确认所使用的替代符没有与正文中的内容冲突，否则正文中的内容将被更改。

必须注意的是在中文 WS 中，不允许只用一个键盘符号作为替代符。因为一个 ASCII 码字符很容易与汉字内码一个字节的高低 7 位相同，造成大量汉字的一半被替换，而使整篇文章变得面目全非。必须使用若干个不与正文内容冲突的 ASCII 码所组成的字符串作为替代符。

另外，如果想加快自动替换的速度，那么在替换进行时打入 GH-C 或 GH-Break 键，则替换过程不再显示出来，将提高替换速度几十倍。

字符数的快速汇总

用 WS 的 D 命令编辑完一篇文章后，很想知道全文有多少字符数，但是 WS 是按页分别统计字符数的，这时只要键入 <Ctrl>+OP 消除页标志后，就会在顶行公告板上看到总行数、字符总数等，注意，此时光标一定要处于文章的最末处。如果是汉字文章，那么将得到的字符总数除以 2 后，得到的就是全文的总字数了。

字符位置的安排

在制表或编辑一个文件时，经常需要将一行字符居中显示，如果用人工数字节计算，然后在插入状态下按空格键或删除键来调整的话，未免太笨。实际上只需将光标移到居中显示的那一行，按 <F4> 键定好标尺宽度后，简单地按一下 <Ctrl>+OC，这行字符就立即按标尺的宽度自动居中显示了。

有时又想让字符左顶格显示，那么将光标移到上一行后，按下 <Ctrl>+MT，就可以达到目的。

<Ctrl>+OF 命令允许你以某一行字符宽度为标尺标准做为文章行的左右位置进行排版，而不必按 <F3>、<F4> 键左右各定义一次位置，使打印出的文章靠右一些，便于装订。

WS 中任务的嵌套执行

有时在编辑状态下急需对另一组文件进行复制或改名，这时不必将文件存盘，退回到主菜单下，只要在编辑状态下键入 <Ctrl>+KO 命令，WS 将显示出同在主菜单下按下 O 命令（拷贝命令）一样的提示，然后输入源盘及文件名、目标盘及文件名，就可以进行文件拷贝，完成后可继续编辑文件。按下 <Ctrl>+KE 命令，其效果同于在主菜单下执行 E 命令。

修改 WS 的退格键和删除键

WS 的退格键不是删除光标左侧的字符，而是将光标左移一格，而删除键不是删除光标所在处的字符，而是删除光标左侧的字符。与习惯了的 DOS 的规定不同，用起来颇为别扭。现将它们修改为 DOS 的规定，即按退格键删除光标左侧字符，按删除键删除光标所在处字符。

```

C>DEBUG WS.COM
-S 0100 FFFE B4 00 CD 16
CS,2427
-U2427
CS,2427 MOV AH,00
CS,2429 INT 16
-A242B
CS,242B JMP 4BE0
CS,242E NOP
-A4BE0
CS,4BE0 CMP AX,0E08

```

```

CS,4BE3 JZ 4BF1
CS,4BE5 CMP AX,5300
CS,4BE8 JZ 4BF7
CS,4BEA OR AL,AL
CS,4BEC JZ 4BF4
CS,4BEE JMP 2426
CS,4BF1 MOV AX,5300
CS,4BF4 JMP 242F
CS,4BF7 MOV AX,2207
CS,4BFA JMP 4BEA
-W
-Q

```

对 WS 编辑的文件内容进行排序

WS 有很强的字处理功能，但没有提供排序功能，实为遗憾。为了对 WS 编辑的文件内容进行排序，可以借助 dBASE III 来进行。

首先由 WS 的 N 命令编辑好一个文件，若是用 D 命令编辑的，先将这个文件转换为标准的文本文件，确认这个文件的后缀为 TXT。然后进入 dBASE III（或 dBASE III PLUS、FoxBASE PLUS），建立一个数据库，其字段类型、长度要与所需处理的文件相匹配。打开这个数据库，使用下面的命令，将文本文件装入该数据库中：

```
APPEND FROM <文件名> SDF
```

再用 SORT 命令对库中记录排序。注意，这时只对含西文的记录进行了正确的排序，而对汉字记录是按 GB2312-80 的排列顺序排序的，既无法实现完全的音序排序，更无法实现汉字的笔画排序，其原因和解决办法请参看本书第二章第一节第九部分中《汉字记录的排序》的介绍。

排序后再用下面的命令，将完成排序的数据库文件转换为标准的文本文件：

```
USE <数据库文件名>
```

```
COPY TO <目标文件名> SDF
```

最后进入 WS，对该文件进行所需要的处理。

WS 命令的加速执行

原版 WS 的使用手册中给出了一条重复命令 <Ctrl>+QQ，它能连续不断地对紧跟在后面的命令或字符起作用，直到按下空格键为止。这条命令在中文 WS 的使用手册中没有列出，但在软件中保留了这一功能。

比如要使屏幕连续向上滚动，直到文件结束，则可按下 ^QQ^C，使之连续删除光标所在处的行；若按 ^QQ^Y，如果 ^QQ 后面不是命令，而是字符的话，则会连续输入该字符，如 ^QQC，则不是连续滚动屏幕，而是连续输入字母“C”。

^QQ 不仅能使 WS 的许多命令连续执行，而且还允许用户调节连续执行的速度，方法是按下字母键上的数字键 0~9，其速度依次减慢。

WS 的假脱机打印

WS 提供了假脱机打印功能，在主菜单中选择 P 命令，输入要打印的文件名并设置一

些参数, 打印机即开始工作, 此时为了不浪费等待打印时间, 你可以在 WS 中同时进行其他工作, 如用 D 或 N 命令进入编辑状态, 输入或修改一篇文章、一段程序, 而不影响打印输出。只是在以 8086/8088 为 CPU 的低时钟频率的微机上, 机器对输入响应有些迟缓。

如果处在编辑状态中, 想不退出打印文件, 就按下 <Ctrl>+KP, 屏幕上会出现如同在主菜单中选择 P 命令后一样的提示, 输入后打印机开始工作, 你也可以继续编辑, WS 会适时提醒换纸和按相应的键继续打印。再按下 <Ctrl>+KP 时, 就如同在主菜单下执行 P 命令后再次按下 P 键, WS 会给出终止或继续打印的选择, 同执行 P 命令一样, 键入 <Ctrl>+U 将暂停打印。

注意: 在打印输出完成前, 不允许执行 X (退出) 或 <Ctrl>+KX (存盘并退出) 命令。

WS 的字块操作

WS 允许在被编辑的文章的某一段的段首键入 ^KB 或 <F7> 键, 在段尾键入 ^KK 或 <F8> 键, 将整段作为一个整体对待, 即字块, 对其进行搬移、拷贝、删除、隐藏、写盘等操作。但由此定义的字块仅仅是块标记间各行的集合, 还不是一个真正意义上的字块实体, 不能将这个块搬移或拷贝到其他行的行尾或行首, 如果那样做了, 只会打乱原有行序, 导致文件报废。

在原版 WS 的使用指南中说明了一个字块操作的列方式命令 ^KN, 中文 WS 保留了这个命令, 但在使用手册中没有给出。在定义了字块后, 按下 ^KN, 这个块就是一个真正的字块实体了, 任你把它搬移或拷贝到哪里, 都不会打乱那里的原有行序。要注意的是, 按下 ^KN 打开列方式后, 字块首标记左边的字符和字块尾标记右边的字符将不会包含在这个块中, 因此不仅要把字块的首、尾标记设在要定义的块的首、末行, 而且要定义在包含了整个块最宽处的地方, 不然就会漏掉一些内容。有了字块的列方式命令, 就可以方便地编辑含有图表的文件了。

^KN 实际上是一个双态转换开关, 首次键入即打开列方式, 再次按下, 就关闭列方式, 恢复成普通的字块。在原版 WS 中, 打开列方式后, WS 将在状态行上给出列方式打开的提示 "Column", 但在中文 WS 中则没有提示, 不过被定义的字块将被转换显示成一些没有实际意义的符号, 再次按下 ^KN, 则又恢复成原有内容, 不会产生破坏作用。由此可以确定被定义的字块是否处于列方式打开状态。只是这样屏幕显示不太美观, 初学者会对此感到茫然。那么就对 WS.COM 进行些修改, 使其处于列方式下, 在状态行上给出 "列方式" 提示。

C>DEBUG WS.COM	AND	AL, 5F
-A2427	CMP	AL, 0D
CS:2427 JMP 49D8	JNZ	49BA
-E242A 90 90 90 90 90	JMP	3CF6
-A3C8E	CMP	AL, 44
CS:3C8E JMP 49B0	JNZ	49C0
CS:3C91 NOP	JMP	49C4
-A49B0	CMP	AL, 4E
CS:49B0 PUSH AX	JNZ	49D3

CMP	[4BD2], AL	CS:4BA0	NOT	BYTE	PTR [4BD3]
JZ	49D0		POP	AX	
MOV	WORD PTR [4BD3], 0000		CALL	4BAD	
MOV	[4BD2], AL		JMP	2426	
POP	AX		NOP		
JMP	3C95		NOP		
NOP			PUSH	AX	
MOV	AH, 00		PUSH	BX	
INT	16		PUSH	DX	
OR	AL, AL		MOV	AH, 03	
JNZ	49E3		MOV	BH, 00	
JMP	242F		INT	10	
CMP	AX, 250B		PUSH	DX	
JZ	4A06		MOV	AH, 02	
CMP	AX, 310E		MOV	BH, 00	
JZ	4A0E		MOV	DX, 0046	
CMP	AX, 314E		INT	10	
JZ	4A0E		MOV	DX, 4BD5	
CMP	AX, 316E		MOV	AH, 09	
JZ	4A0E		INT	21	
CMP	BYTE PTR [4BD4], FF		POP	DX	
JNZ	4A03		MOV	AH, 02	
MOV	BYTE PTR [4BD4], 00		MOV	BH, 00	
JMP	2426		INT	10	
MOV	BYTE PTR [4BD4], FF		POP	DX	
JMP	2426		POP	BX	
CMP	BYTE PTR [4BD4], FF		POP	AX	
JZ	4A18		RET		
JMP	2426		DB	0, 0, 0, "列方式"	
PUSH	AX	-E3456	47		
CMP	BYTE PTR [4BD3], FF	-E0247	46		
JZ	4A25	-E27ED	01		
MOV	AX, D0C1	-E5087	01		
JMP	4A28	-W			
MOV	AX, D0D0	Writing	5500	bytes	
MOV	[4BD5], AX	-Q			
JMP	4BA0				

-A4BA0

WS的再排版

用 WS 输完一篇文章后，如果每个自然段中只有软回车（行末显示“.”），就可以将光标移至首行，键入 <F4> 键设定标尺，然后按下 <Ctrl>+B，WS 就会自动按标尺宽度对文章自动排版。如果在每一行中都是以硬回车结束的话（行末显示“<”），^ B 就不起作用了。但仍能自动排版。

首先将光标移到要排版段的首行，按下 <F5> 键或 <Ctrl>+QA，屏幕提示：“找？”，输入 <Ctrl>+N，屏幕提示：“换为？”，输入一个空格，在屏幕提示“选择？”后，键入回车，或根据实际情况选择输入，最后将光标移到原处，再按下 <Ctrl>+B，文章就根据当前标尺宽度自动排版了。

也可以将整个文件定义为一个字块，并键入`KN，打开列方式，按下`KW，给出新文件名，将文件重新存盘，然后再调入这个文件，将光标移到要排版段的首行，键入`B就能实现排版。因为在字块列方式下，硬回车不随字块转移。

如果先键入`QQ，再键入`B，就能实现对整篇文章的自动排版。

使光标一次移动两个字节

在使用中文 WS 编辑汉字时，光标一次只能移动一个字节，跨在一个汉字的中间，若仅删除半个汉字（一个字节），则会引起汉字内码错位，显示混乱。解决的方法是使光标一次移动两个字节（一个汉字字符）。

```
C>BEBUG WS.COM
-A2427
CS:2427 CALL 4B90
NOP
-A4B90
CS:4B90 PUSH BX
PUSH CX
MOV CX,[4764]
XOR AX,AX
CMP AL,[066B]
JZ 4BDE
MOV [066B],AL
MOV BX,CX
DEC BX
CMP BYTE PTR[BX],A1
JB 4BA0
NOT AH
JMP 4BA3
MOV BX,[4766]
CMP BYTE PTR[BX],A1
JB 4BBB
NOT AL
INC BX
JMP 4BB1
OR AH,AH
JZ 4BD6
OR AL,AL
JZ 4BD2
CMP CX,[0669]
JNB 4BCD
MOV AL,13
JMP 4BCF
MOV AL,04
POP CX

POP BX
RET
MOV AL,7F
JMP 4BCF
OR AL,AL
JZ 4BDE
MOV AL,07
JMP 4BCF
INT 16
MOV [0669],CX
CMP AL,A1
JNB 4BCF
CMP BYTE PTR[463E],00
JNZ 4BCF
CALL 02E0
MOV BYTE PTR[066B],FF
JMP 4BCF

-A02E0
CS:02E0 MOV BX,CX
CMP AH,53
JNZ 02FE
MOV AL,7F
DEC BX
CMP BYTE PTR[BX],A1
JB 02FD
MOV BYTE PTR[240A],01
MOV WORD PTR[240B],066C
MOV [066C],AL
RET
CMP AL,07
JZ 02EA
RET

-W
-Q
```

为中文 WS 增加路径选择

西文 WS4.0 版的各种与文件名有关的命令都允许路径选择，但是中文 WS 是基于 WS3.x 版本汉化的，只允许盘符选择，而不允许路径选择，从而导致了在同一硬盘中的

许多子目录都装有 WS，白白浪费了磁盘空间。对此提出下面两种解决方法。

一、不同路径下共享一份 WS

DOS 提供了一条 PATH 命令，允许在不同路径下，调用 PATH 中指定路径下的 .COM、.EXE 和 .BAT 文件，DOS3.30 版中又增加了一条 APPEND 命令，允许调用其他路径下的除以上三种类型外的其他相关文件。因此将这两条命令联用，就可以在不同的路径下共享一份 WS。

比如中文 WS 装在硬盘的子目录\CWS 中，则执行：

```
PATH C:\CWS
```

```
APPEND C:\CWS
```

以后就可任在何子目录下调用\CWS 中的中文 WS 了。

二、在 WS 中使用路径

DOS3.20 以上版本中有这样一条命令 SUBST，其功能是允许用不同的盘符来查阅其他驱动器或路径，比如执行：

```
SUBST G: C:\MYFILES
```

则盘符 G: 将指向硬盘上的子目录\MYFILES，以后对 G 盘的任何操作，实际均是对子目录\MYFILES 的操作。这里可用的最大盘符取决于 CONFIG.SYS 文件中 LASTDRIVE 语句的规定。

比如在硬盘子目录\CWS 中运行 WS，但编辑的文件想存入硬盘的子目录\MYFILES 中，那么在调用 WS 前执行上述那条 SUBST 命令，然后在用 WS 的 D 或 N 命令输入文件名时，在文件名前加上“G:”，则编辑完后的文件将被存入子目录\MYFILES 中。

修改字型控制

中文 WS 不能全部支持有 16 种字型的打印机驱动程序，原因是没有完整地输出字型控制代码，即 ESC+“1”+“A~P”。下面对此进行修改，使中文 WS 在 CCDOS2.1 下能输出 16 种字型控制代码。

```
C>DEBUG WS.COM
-E0852 91
-E0857 8C
-E08A2 90
-E140D 90 90
-E14F5 90 90
-W
Writing 5500 bytes
-Q
```

相同文件的多份打印

WS 一次只能打印一份文件，怎样才能一次打印出多份相同的文件呢？这里提供两种方法。

1. 如果文件较短，则可以在 WS 的编辑状态下，将整个文件定义为一个字块，然后将光标移至文末，按下 KC，拷贝出一个相同的字块，如此在文末连续拷贝，直到拷贝

数等于想要打印的份数。当然可以把每一块放在新的一页上，便于打印后分发。最后将这个文件打印出来。注意，打印时应用 OP 命令取消页码打印，或参考本节中《常用点命令的使用经验》介绍的方法，在每一块中设置新的页码。

2. 对于很长的文件，虽然也可以采用上面的方法，但最好用 WS 的 P 命令，将其输出到磁盘上，然后调用下面的 BASIC 程序，来完成多份打印。

```
10 CLS:KEY OFF
20 FOR NUM=1 TO n:n 为打印份数
30 PRINT "正在打印，请稍候!"
40 SHELL "TYPE 文件名>PRN" ;文件名由 WS 的 P 命令输出
50 NEXT NUM
60 PRINT "打印结束"
70 END
```

中文 WS 打印文件的行距设置

一些汉化版的 WS 丧失了原版 WS 调节行距的灵活命令，致使在打印文件时很难改变行距。下面给出一个使用 LQ1000K 打印机按不同的行距打印 WS 编辑的文件的 BASIC 程序。

首先用 WS 的 N 命令（或 EDLIN 程序）编辑文件，在需要变化行距处用“ESC3+n”输入设置行距标记，n 为行距设定值，即 $n/180$ 英寸。如 n 取 20、30 时，行距分别为 $1/9$ 、 $1/6$ 英寸。n 的取值范围在 1~255 之间。不足三位时用空格补齐。在需要变换字型处用“ESC1 十字型代号”输入字型变换标记，字型代号取值范围在 CCDOS 字型控制符 A~P 之间。最好采用纯中文状态（全角）输入文件，各行以硬回车结束。

最后调用以下程序，输入相应数据，即可按要求打印。标准页长度为 11 英寸，不论行距怎样变化，只是每一页中行数发生变化，只有当累积长度达到页长时，才会换页。

```
10 CLS:KEY OFF
20 PRINT "请输入以下参数:"
30 INPUT "打印文件名:"; FILE$
40 INPUT "行距[N/180 英寸]:"; LINESPA
50 INPUT "页长[1-22 英寸]:"; PAGELEN
60 INPUT "换页空格:"; SPACE
80 OPEN FILE$ FOR INPUT AS #1
90 LPRINT CHR$(27); "C"; CHR$(0); CHR$(PAGELEN);
100 LPRINT CHR$(27); "N"; CHR$(SPACE);
110 LPRINT CHR$(27); "3"; CHR$(LINESPA);
120 IF EOF(1) THEN CLOSE #1;END
130 LINE INPUT #1;A$
140 IF LEFT$(A$, 4) = "ESC3" THEN LS=VAL(RIGHT$(A$, 3));LPRINT CHR$(27); "3"; CHR$(LS); GOTO 130
150 IF LEFT$(A$, 4) = "ESC1" THEN CT$=RIGHT$(A$, 3);LPRINT CHR$(27); "1"; CT$; ; GOTO 130
160 LPRINT A$
170 GOTO 120
```


文章段落直接打印

如果想打印一篇文章中的一些段落，一般是在 WS 的编辑状态下将这些段落设置成字块，再存入盘中，然后用打印命令将它们一一打印出来，这样操作比较麻烦，还有更为简捷的方法。

在编辑状态下定义好字块后，按下` KW，WS 将要求输入文件名，这时键入“PRN”并回车，这个字块中的内容就会被立即打印出来。这样方法同样适用于 HW 软件。

防止点命令与正文内容混淆

WS 的所有版本都规定一条点命令独占一行，并以“.”作为开始，在打印文本时任何以句点开始的行都被认为是点命令或注释，而不予打印。

然而在编辑、排版正文时，有可能在一行的行首出现作为正文内容的句点，如“.EXE”、“.BAK”等，这样一来，WS 在打印时，该行将被漏丢，使打印出的文件报废。

解决的方法是，在这些行的行首连续按两次` PB，就可以确保正确打印。

常用点命令的使用经验

点命令是 WS 命令中的一个大类，点命令的合理使用，会使许多编辑工作变得方便快捷。下面介绍一些常用点命令的使用经验。

.CPn 强制分页命令

使用这条命令将会插入一个条件分页线，使以后的 n 行保持在同一页上。如果在当前页装不下时，则在此命令行产生分页，以便后面的 n 行完整地保持在下一页上。

这条命令对于打印含有图表的文件特别有用，只要在.CP 后加上图表的行数值，就不必担心图表会跨在分页线上。

.POn 设正文左边空格数 n

这条命令设置页的左边界和第一列文本之间的列数，默认值为 8 列。使用这条命令不仅能打印出的正文靠右一些，以便装订，而且能使由于正文中因字型设置的不同而难以调整各字型文本间相对打印位置的难题，得以迎刃而解。

.. "text" 注释命令

如果想对文本加此注释，则只需在注释前加上“..”就可以了。如果注释超过了一行，就需要在下一行行首加上“..”。注释行中的内容不会被打印出来。

.PLn 设置每页行数为 n

WS 默认每页为 55 行。利用这条命令可以根据实际需要改变页长，n 值可在 1~244 行之间选择。

.LSn (^ OS) 设置行距命令

WordStar4.0 版中增加了.LS 命令，在 4.0 以下版本中同等的命令为` OS，允许用双间距、三间距等方式打印文本。

在.LS 命令后需输入 1~9 中的任一数字，而键入` OS 后，WS 亦要求输入 1~9 中的任一数字。其含义是 1 代表单间距，即行间无空行，2 代表双间距，行间有一空行。3 代表三间距，行间有两个空行，以下类推。

.LHn 设置行高命令

WS 默认行高为 8/48 英寸，即每英寸可输出 6 行文本。LH 命令可以改变这个值。n 取值范围在 2~255 之间，默认值为 8。下表是常用的 n 值与对应的行高

行数/英寸	9.6	8	6.8	6	5.8	4.8	4	3	2
n 值	5	6	7	8	9	10	12	16	24

这条命令将影响每页所能打印的行数。

.PL、.LS(OS)和.LH 命令联合使用，将会大大发挥打印机的控制功能。某些汉字操作系统，如 CCDS2.13 系列也提供了控制打印行距的功能，因此也可以在.PL 命令后紧跟这些控制命令，同样也能达到设置打印文本格式的目的。

.HE“text” 设置上标题注释命令

可以在.HE 后加上一些如文件名、作者、功能、日期等说明，打印文本时，将在每一页的顶部都打印出相同的上标题注释，以便归类文件和防止散乱。

在 WordStar4.0 版中增加了.H1、.H2 和.H3 以标注上标题注释的第一、二、三行。在相同点命令后紧跟回车键，将关闭上标题注释。

.FO 设置下标题注释命令

.FMn 设置每页最后一行正文和第一个下标题注释之间的空行数

.FO 命令的使用同于.HE 命令，只是将下标题打印在每页底部。.HE 和.FO 命令增加的标题不占用每页的正文行数。

WS 默认每页最后一行正文和第一个下标题间的空行数为 2。在.OP 取消页号的打印方式下，.FO 与.FM 命令的联用，可以实现打印“第 n 页”的目的。

.PN 和.PNn 设置打印页号命令

.PN 命令将取消.OP 消除页号的命令，从此页开始打印页号。.PNn 命令将当前页号设置为 n，后续页将以 n 为基准、以 1 为增量连续标注页号。如果前面设置了.OP 命令，.PNn 命令也将取消.OP 禁止页号打印的限制。

.PCn 设置页号位置命令

该命令允许将页标注号打印在不同的列上，其位置取决于 n 值。WS 默认页号打印在第 28 列上。n 值的大小受限于打印机的行宽。该命令对于对折打印和在宽行打印机上打印文本很有用，能将页号标注在合适的位置上。

注意点命令应作为字符串写在文件的空行上，每个命令各占一行。

在 CEGA 卡上使用中文 WS

中文 WS 可以在 GW0520C-H 机上正常运行，但却不能用于 GW286 等配有 CEGA 显示卡的微机上。关键在于 WS.COM 调用 BIOS 的 10H 中断的 04H、06H 功能设置屏幕模式，而在 CEGA 的这种模式中，无法正常显示汉字。现修改如下：

```
C>DEBUG WS.COM
-S0000 FFFE B8 04 00 CD 10
CS:50AD
-E50AE 03
-S0000 FFFE B8 06 00 CD 10
```

10 --

```
-ES0C7 03
-W
Writing 5500 bytes
-Q
```

经修改后 WS 不仅可以在 GW286 等机器中运行，亦可运行于 GW0520C-H 中。

打印竖排汉字

一般打印机都有将字符旋转 90 度打印的功能，但在 WS 下却不能直接调用此功能。调用以下程序后，就可以用 WS 打印出竖排的汉字文本。

```
C>DEBUG
-A
CS:0100 MOV AH,05
CS:0102 MOV DL,1C
CS:0104 INT 21
CS:0106 MOV DL,4A
CS:0108 INT 21
CS:010A MOV AH,4C
CS:010C INT 21
-RCX
CX 0000
:0E
-NUPRIGHT.COM
-W
-Q
```

此程序适用于 OKI8320C 打印机，对于其他打印机，只需修改 CS: 0106 行中的 4A 为相应打印机的旋转打印控制码即可。

用 WS 编辑源程序

WS 提供了诸如字符查找、替换、块操作等功能，强于一般程序设计语言的编辑器。因此用 WS 来编制源程序，可以提高效率。

用 WS 编制源程序时必须使用 N 命令提供的编辑状态，以保证最终形成标准的文本文件，并用文件名中的扩展名必须为相应语言环境承认的扩展名，如 .BAS、.PRG、.PAS 等。如果不慎使用了 D 命令来编制，则请参看本节的有关介绍，将其转换为标准的文本文件。

用 N 命令修改已有的由其他编辑器生成的源文件时，必须保证该文件是标准文本文件。如一般的 BAS 文件是压缩二进制文件，所以必须先在 BASIC 状态中调入要修改的文件，再以 SAVE 加 A 参数存盘，然后用 N 命令调入修改。

中文 WS 在西文状态下的使用

中文 WS 在西文状态下运行时，光标消失，屏幕显示状态不良。下面对 WS.COM 文件进行改造，使之自动判断屏幕是处于西文状态还是中文状态，以便选择不同的显示方式。

```
C>DEBUG WS.COM
-A
CS:0100 JMP 5430
-A5430
```

CS:5430	PUSH	ES	MOV	[50C7], AL
	PUSH	AX	MOV	AX, 000E
	PUSH	BX	MOV	[50C4], AH
	MOV	AH, 0F	MOV	[028B], AL
	INT	10	POP	BX
	CMP	AL, 04	POP	AX
	JGE	543D	POP	ES
	JMP	5441	JMP	3AB1
	CMP	AL, 06	-RCX	
	JBE	545C	CX	5500
	MOV	AX, 1803	,5530	
	MOV	[0248], AH	-W	
	MOV	[508A], AH	-Q	
	MOV	[050C], AL		

扩大 WS 的行宽

WS 对编辑的文本文件限制每行字节数不多于 240 个，超过此数的如一些较大的报表不能用 WS 进行编辑，WS 的这种行宽限制并没有达到常见的 24 点阵打印机的行宽极限值。M2024、M1724 打印机每行为 2176 列，可输出 16 点阵汉字 136 个，8 列字符 272 个，TH3070 打印机每行为 2448 列，可输出 16 点阵汉字 153 个，8 列字符 306 个，9 针打印机 FX-100，采用 4 倍密度位映象方式，每行为 3264 列，可输出 16 点阵汉字 204 个，8 列字符 408 个。因此扩大 WS 的行宽，可以充分利用打印机资源。

下面对 WS 的覆盖文件 WSOVLY1.OVR (长度为 41216 字节) 进行修改，扩大行宽达 274 个字符，可用 272 个字符。这是针对 M1724 打印机的，读者可根据自己的打印机设置相应的行宽值。

首先在 WSOVLY1.OVR 中找到如下两个程序段，对于不同的版本其偏移地址可能会不同。

C> DEBUG WSOVLY1.OVR	CS:26B6	JNB	26BA
-U2983 299A	CS:26B8	MOV	AL, 02
CS:2983 OR AL, DH	CS:26BA	INC	AL
CS:2985 MOV AL, DL	CS:26BC	MOV	[44DF], AL
CS:2987 POP DX	CS:26BF	DEC	AL
CS:2988 JZ 298C	CS:26C1	DEC	AL
CS:298A JMP 2941	CS:26C3	DEC	AL
CS:298C CMP AL, F1	CS:26C5	LEA	BX, [44DB]
CS:298E JB 2992	CS:26C9	CMP	AL, [BX]
CS:2990 JMP 2941	CS:26CB	JB	26CE
CS:2992 OR AL, AL	CS:26CD	RET	
CS:2994 POP BX	CS:26CE	MOV	[BX], AL
CS:2995 POP SI	CS:26D0	LEA	BX, [44DD]
CS:2996 XCHG SI, BX	CS:26D4	CMP	AL, [BX]
CS:2998 PUSH SI	CS:26D6	JB	26D9
CS:2999 POP BX	CS:26D8	RET	
CS:299A RET	CS:26D9	MOV	[BX], AL
-U26B4 26D8	CS:26DA	RET	
CS:26B4 CMP AL, 02			

找到后, 将这两段按以下修改:

```
-A2983
CS:2983  CMP     AL,00
          MOV     AX,DX
          POP     DX
          JZ      298C
          JMP     2941
          CMP     AX,0113 ;行宽最大值加1
          JNB     298A
          OR      AX,AX
          POP     BX
          POP     SI
          XCHG    SI,BX
          PUSH    SI
          POP     BX
          RET

-A26B4
CS:26B4  CMP     AX,0002
          JNB     26BC

          MOV     AX,0002
          INC     AX
          MOV     [44DF],AX
          SUB     AX,0003
          LEA     BX,[44DB]
          CMP     AX,[BX]
          JB      26CC
          RET
          MOV     [BX],AX
          LEA     BX,[44DD]
          CMP     AX,[BX]
          JB      26D7
          RET
          MOV     [BX],AX
          RET
          NOP

-W
-Q
```

修改后, 对 WS 的其他命令没有影响。

文件目录的显示

在 WS 编辑状态下编辑文章时, 突然想看一看当前路径中有些什么文件, 怎么办? 退出吗? 大可不必, 按下 <Ctrl>+KF 命令, 看看什么事发生。这时就象在 DOS 状态下键入命令 DIR /W 一样, 当前路径下的文件以紧凑方式给你列了出来, 一列显五个文件名, 包括扩展名, 再按一次 <Ctrl>+KF 命令, 就回到了编辑状态。

如果你突然忘了某条命令, 那就按下 <Ctrl>+J 键吧, WS 会立刻帮助你的。

文件不能存盘的解决方法

在用 WS 编辑文件时, 偶尔会由于磁盘空间满, 不能将文件存盘的情况, 这时千万不要轻易放弃, 只要还在 WS 的控制下, 就可以采用以下方法将文件再次存入盘中。

1. 将整个文件定义为一个字块, 换一张空间足够的磁盘, 键入写文件命令 <Ctrl>+KW, 将整个块写入一个新命令的文件中。也可以将一篇长文章分成若干块, 用 KW 命令分别写到几个文件中, 然后用 DOS 的 COPY 命令来链接, 或在 WS 中打开一个文件, 用 KR 命令依次读入另外几个文件, 实现链接。

2. 如果没有可用的其他磁盘, 则用 KJ 命令删去当前路径下的一些文件, 如果不清楚可删去的文件名, 则键入 KF, 然后按 ^Z, ^W 上下查看, 如果要删除的文件不在当前路径中, 就用 KL 命令来改变路径, 重新查看。文件被删除后, 就可以将定义的块用 ^KW 写入磁盘中了。

3. 在运行 WS 前, 先将 PCTOOLS 驱留内存, 一旦出现不能存盘的现象, 立即按下热键 <Alt>+<Esc>, 调出 PCTOOLS, 完成查看文件、删除文件操作, 再返回 WS, 将文件以 ^KW 存盘。

万一未将文件存盘, 又不慎退出了 WS, 那么只好用 DEBUG 来从内存中恢复文件

了。

```
C>DEBUG
-S0000 FFFE 'String1'
CS:offset1
-S0000 FFFE 'String2'
CS:offset2
-H offset2 offset1
x1 x2
-RCX
CX 00 00
; X2+1
-N <FILENAME>
-W OFFSET1
-Q
```

其中 String1 和 String2 是未存盘文件的首、尾几个字符，offset1 和 offset2 是查找到的地址值，可能会有好几个，这时要用 D 命令查看一下内容，确定正确的值，X2 是 offset2 与 offset1 的差，即丢失的文件的大致长度，filename 是给定的文件名。若文件长度超过 FFFF，还必须将其高 4 位写入 BX 寄存器。

经过以上努力，丢失的文件就基本被恢复了，再进入 WS 对其进行整理、修改。

在以上操作过程中，如果遇上系统死锁或被重置，那只好从头开始了。

中西文 WS 的文件转换

由西文 WS 编辑的文件不能在中文 WS 中直接使用。为了使西文 WS 编辑的英文文章经中文 WS 编辑后，也能在某些汉字系统的支持下，打印出如中空、斜体等特殊字型，就必须将文件经过转换。下面给出一段将西文 WS 编辑的文件转为中文 WS 可用文件的 BASIC 程序。

```
5 DEFINT H-N,DIM B$(500),CLS,DEY OFF
10 INPUT "Input SOURCE file name: ",SF$
20 INPUT "Input TARGET file name: ",TF$
30 OPEN SF$ FOR INPUT AS #1
40 QO$="" :KO=0:N=0
50 LINE INPUT #1,Q$:IF EOF(1) THEN 180
60 IF Q$="" THEN N=N+1:B$(N)="" :PRINT N;" " :B$(N):KO=0:GOTO 50
70 IF LEN(Q$)=255 THEN K1=1 ELSE K1=0
80 H=INSTR(Q$,".PA"):IF H>0 THEN Q$=LEFT$(Q$,H-1)+RIGHT$(Q$,LEN(Q$)-H-2)
90 H=INSTR(Q$,".pa"):IF H>0 THEN Q$=LEFT$(Q$,H-1)+RIGHT$(Q$,LEN(Q$)-H-2)
100 IF KO=0 THEN K2=0:GOTO 130
110 H=LEN(QO$):L=LEN(Q$):IF H+L<=255 THEN Q$=QO$+Q$:K2=0:GOTO 130
120 Y$=RIGHT$(Q$,L+H-255):Q$=QO$+LEFT$(Q$,255-H):K2=1
130 H=INSTR(Q$,CHR$(141)):IF H=0 THEN 150
140 N=N+1:B$(N)=LEFT$(Q$,H-2):Q$=RIGHT$(Q$,LEN(Q$)-H-1):GOSUB 210:GOTO 130
150 IF K2>0 THEN Q$=Q$+Y$:K2=0:GOTO 130
160 IF K1=0 THEN KO=0:N=N+1:B$(N)=Q$:GOSUB 210:GOTO 50
170 KO=1:QO$=Q$:GOTO 50
180 CLOSE #1:OPEN TF$ FOR OUTPUT AS #1
190 FOR I=1 TO N:PRINT #1,B$(I):NEXT I:CLOSE #1
200 END
```

```

210 H=0;L=LEN(B$(N))
220 H=H+1;IF H>L THEN 250
230 M=ASC(MID$(B$(N),H,1));IF M<128 THEN 220
240 B$(N)=MID$(B$(N),1,H-1)+CHR$(M-128)+MID$(B$(N),H+1,L-H);GOTO 220
250 PRINT N;" ";B$(N);RETURN

```

将 WS 的文件转换为标准文本文件

用 WS 的 D 命令编辑的文件不是标准的文本文件，其中含有软回车、分页符等控制符，不能为 DOS 或其他软件接受。下面介绍几个将其转换为标准文本文件的程序。

1. 用 WS 的 N 命令调入要转换的文件，在每行末加上硬回车，然后删除分页处的几行（WS 默认 55 行为一页），再重新输入这几行并重新存盘后，就是标准文本文件了。这种方法对于转换较大的文件就不适用了。

2. 程序转换法

程序一

```

C>DEBUG
-A
CS:0100  MOV  AH,08
          INT  21
          AND  AL,7F
          MOV  DL,AL
          MOV  AH,02
          INT  21
          CMP  AL,1A
          JNZ  100
          MOV  AH,4C
          INT  21

-RCX
CX  0000
:14
-NTRAN.COM
-W
-Q

```

使用方法：

C>TRAN<源文件名>目标文件名。

程序二

这是用 Turbo PASCAL3.0 以上版编制的。

```

PROGRAM TRAN (f1,f2);
VAR
    f1,f2:text;
    chu:char;
BEGIN
    reset(f1);
    rewrite(f2);
    WHILE not (eof(f1)) DO
        BEGIN

```

```

        IF eof(f1) THEN
        BEGIN
            readln (f1);
            writeln (f2);
        END;
        ELSE
        BEGIN
            read (f1, ch);
            write (f2, chr (asc(ch) AND 127));
        END;
    END;
END.

```

使用方法:

C>TRAN 源文件名 目标文件名

程序三

这是用 GWBASIC 编制的。

```

50 CLEAR
60 OPTION BASE 1
70 DIM WS$(101), PT$(101)
80 ON ERROR GOTO 290
90 CLS
110 LOCATE 2,20:INPUT "原文件名:", WSM$
120 LOCATE 3,20:INPUT "目标文件名:", PTM$
130 LOCATE 5,23:PRINT "正在转换!"
140 OPEN WSM$ FOR INPUT AS #1
150 K=1;G=3500
155 LOCATE 7,25:PRINT "共",LOCATE 7,33:PRINT "行"
160 FOR I=1 TO G
164 LOCATE 7,28:PRINT I
170 IF EOF(1) THEN 250
180 LINE INPUT #1, WS$(I); PT$(I)=WS$(I)
190 HC=LEN(WS$(I));FOR J=1 TO HC
200 X$=MID$(WS$(I), J, 1);A=ASC(X$)
210 IF A=141 THEN X$=CHR$(13);MID$(PT$(I), J, 1)=X$
220 IF A=138 THEN X$=CHR$(10);MID$(PT$(I), J, 1)=X$
230 NEXT J
240 K=K+1;NEXT I
250 OPEN "0", #2, PTM$;FOR I=1 TO K:PRINT #2, PT$(I);NEXT I;CLOSE #2
260 CLOSE(1);LOCATE 9,24:PRINT "转换完毕!";BEEP
270 END
290 IF ERR=53 THEN BEEP;CLS;LOCATE 5,30:PRINT "文件",WSM$,"不存在!"
300 LOCATE 7,22:PRINT "按回车键重输(N键退出)"
310 LOCATE 7,55:INPUT " ", P$
320 IF P$="N" OR P$="n" THEN RESUME 270 ELSE RESUME 90

```

为 WS 增加稿纸打印格式

下面是一个用 Turbo PASCAL3.1 以上版本编制的稿纸打印程序，它能按 20×20 的

方格纸形式打印出文本文件。

这个程序可以用 WS 的 R 命令调用，也可以直接嵌入到 WS 的主菜单中，嵌入方法如下：

```
C>DEBUG WS.COM
-A3C14
CS:3C14 JMP 5602
-A5602
CS:5602 PUSH AX
      PUSH BX
      MOV AL,[463E]
      CMP AL,77
      JZ 5613
      POP BX
      POP AX
      MOV AL,[45CB]
      JMP 3C17
      MOV BX,1000
      MOV AH,48
      INT 21
      MOV AL,0
      MOV AH,4B
      MOV DX,[5625]
      INT 21
      JMP 560B
      DB 8,"WSPRT.COM",13
      -RCX
      CX 5500
      5630
      -W
      -Q
```

```
program wsprt;
{ $ U+}
const
  black=' ' ;
  sbblack=' ' ;
var
  Filevar;text;
  ok:boolean;
  buff,line:string[60];
  bpage, page,mpage,i,j,k:integer;
  filename:string[14];
  a:array [1..19] of integer;
  procedure prec;
  var
    n,h,m,j,k: integer;
    tj1,tj2:boolean;
  begin
    j:=1;
    for k:=1 to 19 do
      a[k]:=1;
      k:=1;
      while j<20 do
        begin
          tj1:=ord(line[k])>128;
          tj2:=ord(line[k])=32;
          if tj1 or tj2
          then begin
            a[j]:=1;
            k:=k+2;
            j:=j+1;
          end
          else
            n:=0;
            m:=k;
            tj1:=ord(line[m])<128;
            tj2:=ord(line[m])>32;
            while tj1 and tj2 do
              begin
                n:=n+1;
                m:=m+1;
                tj1:=ord(line [m])<128;
                tj2:=ord(line [m])>=32;
                if ord(line[m])=32 then if ord(line[m-1])=32 then
                  begin
                    tj2:=false;
                    n:=n-1;
                    m:=m-2;
                  end;
                end;
                h:=1;
                tj1:=n<=2;
                while not tj1 do
                  begin
                    h:=h+1;
                    tj1:=n<=2+(h-1)*4;
                  end;
                case h of
                  1,begin
                    a[j]:=1;
                    j:=j+1;
                  end;
                  2,begin
```