

CCDOS V4.0

程序结构注释

中科院希望高级电脑技术公司
《计算机科学技术与应用》编辑部

CC-DOS V4.0

程 序 结 构 与 注 释

钱培德 著

《计算机科学技术与应用》编辑部
中国科学院希望电脑高技术公司

前 言

CC—DOS汉字操作系统自一九八三年问世以来，获得了很大的成功。由于它的强大功能和友好的用户界面，它已成为拥有最多用户的汉字操作系统。并且，它的结构已成为微机汉字操作系统的一种事实上的标准。

CC—DOS已经历了多个版本，这些版本充分体现了CC—DOS的发展过程，展示了一个为人试用的系统发展到一个国外广为采用的系统的历程。CC—DOS的道路并没有走完，它仍在向前发展，CC—DOS V4.0就是它的最新版本。

CC—DOS V4.0集CC—DOS历届版本之所长，并吸收许多CC—DOS变种的优点，再加以充分发挥。所以，这个版本与其先前版本有明显的区别，有人称之为CC—DOS更新换代的产品。正因为如此，许多人对CC—DOS V4.0格外重视，企图学习它的先进技术和实现方法。令人遗憾的是，它的设计者为了维护该软件的合法版权和经济利益，在系统中采用了加密和保密技术，从而给对这个系统的分析设置了障碍。许多专业人员采用传统的方法来分析CC—DOS V4.0，均遭到了失败。

本人对CC—DOS V4.0具有极大的兴趣，在经过一段时间的奋斗后，终于使其“真相大白”。本人通过对它进行分析后觉得，CC—DOS V4.0设计新颖、独特、严谨、完善和先进，很值得大家一学。故特撰写成书，供广大计算机系统专业人员学习参考。

本书中提供了较完整的CC—DOS V4.0程序清单，并把这些程序按结构进行了划分。每个子程序前给出了功能介绍和必要的入口参数及出口参数的说明。并且基本上给每条指令加了简要说明。这样，读者较易看清CC—DOS V4.0的结构，以及各部分的作用和功能。

值得指出的是，本书中给出的程序清单不是源程序，而是先对CC—DOS V4.0的目标文件代码进行解密复原，再用DEBUG进行反汇编得到的。为了清楚起见，书中把数据区和工作区内容均用伪指令DB和DW等表示。另外，为节省篇幅，对那些较大的数据区（如汉字库和对照表等），只给出其地址，而省略其内容。

本书的形成过程中得到了金传祚高级工程师的大力支持，杨季文同志为本书作了注释信息的输入工作，本人谨在此向他们深表谢意。

关于CC—DOS V4.0的分析报告，本人将另行撰写成书。

钱培德

1989年7月于苏州大学计算机工程系

内 容 简 介

本书提供了CC—DOS V4.0的程序清单，并按结构作了分段，给指令加了简要的注释。程序清单中包括自举模块、键盘管理模块、显示器控制模块和打印机驱动模块，以及各个数据区和工作区，本书是学习CC—DOS V4.0的必备参考书，也可作为有关专业课与培训班的教材和参考书。

目 录

- 第一章 系统总述
- 第二章 CCCC·COM文件
- 第三章 CCCC·OVR文件
- 第四章 CCCC·OV 1 文件
- 第五章 M2024E·COM文件
- 第六章 M2024P·COM文件

第一章 系统总述

一、系统文件

CC—DOS V4.0是CC—DOS历届版本中拥有最多系统磁盘的版本，它存放在五张360K字节的软磁盘上。这些盘上存放的文件之作用如下所述。

1号磁盘上有下列文件：

CLIB241	24点阵字库第一部分
LOAD.BAT	装配24点阵字库命令

2号磁盘上有下列文件：

CLIB242	24点阵字库第二部分
LOAD.BAT	装配24点阵字库命令

3号磁盘上有下列文件：

CCCC.COM	汉字系统装入文件
CCCC.OVR	键盘管理模块
CCCC.OV1	彩色显示器控制模块
CCCC.OV2	单色显示器控制模块
CCLIB	16点阵字库
CH.EXE	高频字统计和改码表文件
CH.MSG	记录码表文件

4号磁盘上有下列文件：

TH1351E.COM	TH1351的24点阵打印程序
1351P.COM	TH1351的16点阵打印程序
TH3070E.COM	TH3070的24点阵打印程序
3070P.COM	TH3070的16点阵打印程序
M2024E.COM	M2024的24点阵打印程序
M2024P.COM	M2024的16点阵打印程序
NECP7E.COM	NECP7的24点阵打印程序
NECP7P.COM	NECP7的16点阵打印程序
NEC9400E.COM	NEC9400的24点阵打印程序
9400P.COM	NEC9400的16点阵打印程序
9P.EXE	9针打印机的打印程序

5号磁盘上有下列文件：

CH16.EXE	16点阵造字程序
CH24.EXE	24点阵造字程序
LOADCZ.EXE	装入词组库程序
CZ.EXE	造词库程序

KEYDZ.COM	大众输入方法
KEYWB.COM	五笔输入方法
KEYCJ.COM	仓颉输入方法
KEYDB.COM	电报输入方法
KEYPX.COM	拼形输入方法
KEYGY.COM	广东话拼音输入方法
KEYSS.COM	声韵、声声输入法装入命令
KEYSS.OV1	声声模块
KEYSS.OV2	声韵模块
KEYBX.COM	笔形输入方法

以上这些CC—DOS V4.0的系统文件中最主要的文件为汉字系统装入文件(CCCC.COM)、键盘管理模块(CCCC.OVR)、显示器控制模块(CCCC.OV1或CCCC.OV2)和打印机驱动程序(例如M2024E.COM和M2024P.COM)。本书的后面几章将给出这些重要文件的清单及注释。

CC—DOS V4.0也要基于西文DOS的核心运行,它需要DOS的版本号不低于2.0版。

二、系统功能

CC—DOS V4.0具有完备的功能,它除了具有一般汉字操作系统均有的汉字I/O功能外,还具有其独特的功能。

CC—DOS V4.0的汉字库装入,可由用户选择决定,以让用户自己调节系统占用的内存量。系统自举时,会与用户进行汉字库装入选择的人机对话,用户可以要求把汉字库全部装入内存,也可以只装入一级汉字,也可以全部不装入内存,甚至可以任意切割汉字库装入内存。当用户选择部分汉字库装入内存时,剩余部分则驻留在磁盘上,若要用到这些内容,系统中具有专门程序,采用BIOS中断实现对它们的较快速访问,并且不会产生DOS中断的重入问题。在汉字库全部不装入内存时,整个系统所占用的内存量仅40K字节左右。

CC—DOS V4.0的系统盘采用了激光加密技术,因此具有可靠的反拷贝能力。为了防止别人擅自解密,设计者对加密部分采用了多种保密措施,有效地维护了该系统的合法版权。

CC—DOS V4.0拥有一整套汉字输入方式,这些输入方式均是从我国的500余种汉字输入方式中挑选出来的,它里面有音码、形码、形音码和流水码。这些输入方式中有普及型的,也有专用型的,故可以覆盖各种用户的需求。系统中还留有一个增加新的汉字输入方式的软接口,系统还提供了一系列附加的汉字输入方式处理程序,这些程序就是通过这个接口联接到系统上的。

系统中增加了字典功能,使用该功能后,当用一种编码输入汉字时,这个汉字的其它几种输入方式的对应编码也会出现在提示行内。这样有利于用户学习有关的汉字输入方法。

为了加快汉字的输入速度,系统具有高频字统计功能。使用这个功能后,系统能自动对当前的汉字使用情况进行统计,并自动定义出10个一键输入常用字和10个二键输入常用字。当用户打入这些字的编码时,常用字在提示行中首先出现,加快了输入速度。

为了增加用户使用的灵活性,系统允许用户在西文方式和中文方式间任意切换。值得一

提的是,当由中文方式切换到西文方式时,系统将把中文方式多占的内存空间全部释放出来,实现了名副其实地回到西文状态。

系统提供了动态修改对照表的功能,用户可以随时对对照表作出修改。这种修改的界面十分友好,采用自然的人机会话,大大方便了用户。

为了方便用户在汉字输入方式下输入图形符号和制表符,系统增加了图形符号功能和制表符功能。采用这两种功能后,用户可以在任何汉字输入方式下通过定义的数字键输入常用图形符,还可通过右边的小键盘输入制表符。从而大大减少了切换输入方式的次数,提高了输入速度。

CC-DOS V4.0的显示模块具有较广的适用范围,它不但能适用于CGA、EGA和CGE等彩色显示系统,而且还能适用于HGC和MGA等单色显示系统。系统根据配置的显示适配卡自动控制装入内存的显示模块。

为了增强系统与西文DOS的兼容性,显示模块中的滚屏和TTY方式显示部分均作了加强,从而可使该系统与西文DOS在显示方面高度兼容。

CC-DOS V4.0具有完备的汉字打印功能。系统提供了各种常用打印机的打印驱动程序,从而可以支持各种常用的打印机。它不但能输出 16×16 点阵的汉字,而且可以输出较高质量的 24×24 点阵的汉字,并可分别打印输出8种和14种字型。

系统向用户开放了打印机固有的全部控制命令,用户在程序中可以任意使用各种控制命令。系统还向用户提供了重定义字型、行宽和字间距的功能,用户可以用键盘命令或在程序中发命令,来改变打印机的输出字型、输出行宽和字间距。为了方便用户打印各种表格,系统还允许用户用键盘或程序来改变行间距。

系统具有完备的屏幕硬拷贝功能,这种硬拷贝可适用于各种显示系统。硬拷贝产生的字型也可改变,可满足用户的不同要求。

CC-DOS V4.0还能适用于配有国产CEGA卡的系统。它能自动检测出CEGA卡的存在,并自动对系统的自举进行调整,以充分利用CEGA卡的资源(汉字库和分辨率等)。CC-DOS V4.0与GW-BIOS具有较好的兼容性。

CC-DOS V4.0还设计有专门的汉卡,系统可以支持汉卡获得优异的系统性能。系统在自举时可对汉卡的存在进行检测,然后自动完成相应的模块装配。

三、系统代码体系

为实现与CC-DOS历届各版本的兼容,CC-DOS V4.0中的汉字机内码仍采用双字节的变形国标码(即国标码高位置1)。字符的机内码采用不完整的扩展ASCII码,其范围为 $00H-A0H$ 。 $00H-7FH$ 表示普通ASCII字符, $80H-A0H$ 表示扩展ASCII字符。 $A1H-FFH$ 为汉字机内码占用,故无法表示这一范围的扩展ASCII字符。

为了实现汉字的输出,故系统内具有汉字字形码(即字模)。CC-DOS V4.0的汉字字形码有两种:一种是表示 16×16 点阵汉字的字形码,码长32字节,字形码与横向点阵相对应,适用于汉字的显示;另一种是表示 24×24 点阵汉字的字形码,码长72字节,字形码与纵向点阵相对应,适用于汉字的打印。

CC-DOS V4.0拥有较多的汉字输入码,它们有区位码、首尾码、拼音码、快速码、大众码、五笔字型、仓颉码、广东拼音码、电报码、拼形码、笔形码、声韵和声声码等。

四、系统自举

当对系统进行冷启动或热启动后,即实现了西文DOS的自举。这时只要从键盘输入以下命令,即可实现CC—DOS V4.0的自举。

CCCC <词库文件名>

如果不需要装入词库,则CCCC后的命令参数可以省略。这样就启动了可执行文件CCC·COM,来完成系统自举。

CCCC·COM纯粹是一个系统装入文件,它首先把程序的内存映象向高地址移动64K字节,以空出一个区域,为以后装入CCCC·OVR文件作好准备。然后打开CCCC·OVR文件,对其有关内容进行复原,再把这个文件读入内存,并完成重定位,使CCCC·OVR在内存中形成可执行文件映象,最后把控制权交给CCCC·OVR,启动它执行。

CCCC·OVR文件的绝大部分内容为CC—DOS V4.0的键盘管理模块,有一部分为自举部分,在自举时执行的只是这一部分内容。键盘管理模块随CCCC·OVR带入内存,这部分内容在自举时没有被执行。CCCC·OVR的自举部分继续完成CCCC·COM所没有完成的自举工作。

首先检查DOS的版本号是否在2.0版以上,再检查命令行中是否有“词库文件名”,如有,则把本程序向高地址方向移动,空出能容下词库的空间,把词库文件内容读入该空间。接着修改DOS内核中的有关滤符指令,使汉字机内码在系统内能正常存在。又修改了16H号中断向量,使它指向新的16H号中断处理程序。然后根据系统配置的显示适配卡选定显示器控制模块文件,当为彩色卡时,则选定CCCC·OV1文件,当为单色卡时,则选定CCCC·OV2文件,当为CEGA卡时,则自举结束。最后,把CCCC·OVR的自举部分向高地址方向移动,空出区域来存放选定的显示器控制模块,接着读入显示器控制模块,复原该模块的内容,把控制交给该模块。

显示器控制模块中的开头一小部分是自举部分,自举时就是执行这部分内容,先是检测系统中是否有汉卡,如没有装汉卡,就把本程序向高地址方向移动,空出能容下所要装入字库的空间,再把汉字库内容读入该区,并建汉字库文件的定位表,以便实现用BIOS中断访问外存字库。然后修改10H号中断向量,使其指向新的10H号中断处理程序。最后计算出驻留内存的程序总长度,并把它们驻留在内存中。自举到此基本结束。

最后还要进行打印驱动模块的自举,这个自举过程较简单,主要是修改5H号中断向量和17H号中断向量,再是把程序驻留内存。要说明的是,24点阵的打印驱动模块自举中,还必须建立24点阵汉字库文件的定位表,以实现今后对该字库的访问。

总的来说,CC—DOS V4.0的自举过程,要比其以前版本的自举复杂得多。

第二章 CCC·COM文件

```

0FE6:0100 JMP 030C ; 转030CH处执行
;
0FE6:0145 DB 'CCCC.OVR',0 ; 文件名存放区
0FE6:014E DW 0 ; 空出区地址 (段址形式)
; 存放单元
0FE6:0154 DW 0 ; 文件柄保存单元
0FE6:0156 DB 20H DUP(0) ; 文件首部缓冲区
0FE6:0176 DB 09H,23H,65H,72H,ACH ; 复原操作数据
DB 29H,DFH,AAH,67H,55H
DB EDH,78H,91H,ABH,73H
DB 5AH,ACH,82H
;
; *****
; * 字符复原子程序 1 *
; *
; * 入口参数: DX = 处理字符的首址 *
; *****
;
0FE6:02BF PUSH BX ; 保存BX
0FE6:02C0 PUSH DS ; DS值入栈
0FE6:02C1 POP ES ; DS值送ES
0FE6:02C2 MOV SI,DX ; 字符首址送SI
0FE6:02C4 MOV DI,SI ; SI值送DI
0FE6:02C6 MOV CX,0004 ; 字节数4送CX
0FE6:02C9 LODSB ; 取一字节到AL
0FE6:02CA XOR BH,BH ; BH清0
0FE6:02CC MOV BL,[0176] ; 复原操作数据送BL
0FE6:02D0 NOT AL ; 字符值取反
0FE6:02D2 SUB AL,BL ; 字符值减去BL值
0FE6:02D4 XOR AL,[BX+0177] ; 字符值异或复原操作数据
;
0FE6:02D8 ADD AL,BL ; 字符值加上BL值
0FE6:02DA STOSB ; 字符存入原处
0FE6:02DB DEC BYTE PTR [0176] ; 复原操作数据减去1
0FE6:02DF JNS 02E6 ; 非负值则转02E6H

```

0FE6:02E1	MOV BYTE PTR [0176],0F	; 复原操作数据置为0FH
0FE6:02E6	LOOP 02C9	; 循环转02C9H执行CX次
0FE6:02E8	POP BX	; 恢复BX
0FE6:02E9	RET	; 返回

```

;
; *****
; *                                     *
; *           字符复原子程序 2           *
; *                                     *
; *   入口参数: DX = 处理字符的首址     *
; *           CX = 字符的个数           *
; *                                     *
; *****
;

```

0FE6:02EA	PUSH BX	; BX值入栈
0FE6:02EB	MOV SI,DX	; 字符首址送SI
0FE6:02ED	MOV DH,0F	; 0FH送DH
0FE6:02EF	MOV DI,SI	; 字符首址送DI
0FE6:02F1	LODSB	; 取一字符送AL
0FE6:02F2	XOR BH,BH	; BH清0
0FE6:02F4	MOV BL,DH	; 0FH送BL
0FE6:02F6	NOT AL	; 字符值取反
0FE6:02F8	SUB AL,BL	; 字符值减去BL值
0FE6:02FA	CS:	
0FE6:02FB	XOR AL,[BX+0177]	; 字符值异或上复原操作数据
0FE6:02FF	ADD AL,BL	; 字符值加上BL值
0FE6:0301	STOSB	; 把字符值存入原处
0FE6:0302	DEC DH	; DH值减去1
0FE6:0304	JNS 0308	; 不为负的则转0308H
0FE6:0306	MOV DH,0F	; 0FH送DH
0FE6:0308	LOOP 02F1	; 循环执行02F1H处
0FE6:030A	POP BX	; 栈顶内容送BX
0FE6:030B	RET	; 返回
0FE6:030C	MOV AX,CS	; 代码段值送入AX
0FE6:030E	ADD AX,1000	; 加上1000H
0FE6:0311	MOV ES,AX	; ES指向64K字节以后的空间
0FE6:0313	LEA CX,[08AC]	; 字节计数08ACH送CX
0FE6:0317	XOR SI,SI	; 源指针清0
0FE6:0319	XOR DI,DI	; 目标指针清0

0FE6 : 031B	PUSH CS	； 代码段值入栈
0FE6 : 031C	POP DS	； 代码段值送DS
0FE6 : 031D	REPZ	； 把程序代码的前08ACH
0FE6 : 031E	MOVSB	； 个字节移到64K字节后
		； 的空间中，空出64K空间
0FE6 : 031F	PUSH ES	； ES值入栈
0FE6 : 0320	LEA AX,[0326]	； 0326H送AX
0FE6 : 0324	PUSH AX	； AX值入栈
0FE6 : 0325	RETF	； 转ES : 0326H处执行
0FE6 : 0326	PUSH CS	； 代码段值入栈
0FE6 : 0327	POP SS	； 代码段值送SS
0FE6 : 0328	MOV AX,DS	； 数据段值送AX
0FE6 : 032A	ADD AX,0010	； AX加上10H，指向原启动点
；		
0FE6 : 032D	CS :	
0FE6 : 032E	MOV [014E],AX	； AX值保存到014EH字单元
；		
0FE6 : 0331	MOV AX,CS	； 代码段值送AX
0FE6 : 0333	MOX DS,AX	； 代码段值送DS
0FE6 : 0335	MOV ES,AX	； 代码段值送ES
0FE6 : 0337	MOV DX,0145	； 文件名串地址送DX
0FE6 : 033A	MOV AX,3D02	； 调用号和工作方式号送AX
；		
0FE6 : 033D	INT 21	； 打开文件CCCC.OVR
0FE6 : 033F	JNB 0344	； 不出错则转0344H
0FE6 : 0341	JMP 04BD	； 出错则转04BDH
0FE6 : 0344	MOV [0154],AX	； 文件柄存入0154H字单元
；		
0FE6 : 0347	MOV BX,AX	； 文件柄送BX
0FE6 : 0349	MOV DX,0156	； 缓冲区地址送DX
0FE6 : 034C	MOV AH,3F	； 调用号送AH
0FE6 : 034E	MOV CX,0020	； 字符个数送CX
0FE6 : 0351	INT 21	； 读入CCCC.OVR的前20H
		； 个字节
0FE6 : 0353	JNB 0358	； 不出错则转0358H
0FE6 : 0355	JMP 04BD	； 出错则转04BDH
0FE6 : 0358	MOV CX,0020	； 字节个数送CX
0FE6 : 035B	CALL 02EA	； 把缓冲区开头的CX个字
		； 节复原
0FE6 : 035E	LEA SI,[0156]	； 缓冲区地址送SI

0FE6 : 0362	CMP	WORD PTR [SI] ,5A4D	；首字为LINK的EXE签字？
0FE6 : 0366	JZ	036B	；为EXE文件则转036BH
0FE6 : 0368	JMP	045C	；转045CH
0FE5 : 036B	MOV	AX, [SI + 08]	；文件首部的节数送AX
0FE6 : 036E	MOV	CL,04	；04H送CL
0FE6 : 0370	SHL	AX,CL	；AX左移4位，得文件首部 ；的字节数
0FE6 : 0372	MOV	DX,AX	；文件正文起点地址送DX
0FE6 : 0374	XOR	CX,CX	；CX清0
0FE6 : 0376	MOV	AX,4200	；调用号和工作方式号送AX
0FE6 : 0379	INT	21	；把CCCC.OVR指针移至文 ；件正文首
0FE6 : 037B	JNB	0380	；不出错则转0380H
0FE6 : 037D	JMP	04BD	；出错则转04BDH
0FE6 : 0380	MOV	DX, [014E]	；空出区首址送DX
0FE6 : 0384	XOR	SI,SI	；SI清0
0FE6 : 0386	MOV	AH,3F	；调用号送AH
0FE6 : 0388	MOV	CX,8000	；字节数8000H送CX
0FE6 : 038B	MOV	DS,DX	；DX值送DS
0FE6 : 038D	XOR	DX,DX	；DX清0
0FE6 : 038F	INT	21	；读CCCC.OVR的8000H个 ；字节入空出区
0FE6 : 0391	JNB	0396	；不出错则转0396H
0FE6 : 0393	JNP	04BD	；出错则转04BDH
0FE6 : 0396	ADD	SI,AX	；SI加入本次实际读入的 ；字节数
0FE6 : 0398	CMP	AX,8000	；本次实际读入字节数与 ；8000H比较
0FE6 : 039B	JB	03A5	；小于则转03A5H
0FE6 : 039D	MOV	DX,DS	；DS值送DX
0FE6 : 039F	ADD	DX,0800	；DX加上0800H
0FE6 : 03A3	JMP	0386	；转0386H
0FE6 : 03A5	CS :		
0FE6 : 03A6	MOV	DS, [014E]	；空出区地址送DS
0FE6 : 03AA	PUSH	DS	；DS值入栈
0FE6 : 03AB	POP	ES	；DS值送ES
0FE6 : 03AC	MOV	CX,SI	；CCCC.OVR读入总字节数 ；送CX
0FE6 : 03AE	MOV	DX,6DC0	；6DC0H送DX

0FE6:03B1	SUB CX,DX	, 总字节数减去6DC0H,得
		, 到要复原的字节数
0FE6:03B3	CALL 02EA	, 把6DC0H偏移量后的内
		, 容复原
0FE6:03B6	PUSH CS	, CS值入栈
0FE6:03B7	POP DS	, CS值送DS
0FE6:03B8	MOV AX,4200	, 调用号和方式号送AX
0FE6:03BB	XOR CX,CX	, CX清0
0FE6:03BD	MOV DX,001E	, 001EH送DX
0FE6:03C0	INT 21	, 把CCCC.OVR指针移至
		, 001EH处
0FE6:03C2	JNB 03C7	, 不出错则转03C7H
0FE6:03C4	JMP 04BD	, 出错则转04BDH
0FE6:03C7	MOV BYTE PTR [0176],01	, 复原操作数据置初值
0FE6:03CC	MOV AH,3F	, 调用号送AH
0FE6:03CE	MOV CX,0004	, 字节数送CX
0FE6:03D1	LEA DX,[0150]	, 缓冲区首址送DX
0FE6:03D5	MOV SI,DX	, DX值送SI
0FE6:03D7	INT 21	, 读文件001EH开始的4字
		, 节到缓冲区
0FE6:03D9	JNB 03DE	, 不出错则转03DEH
0FE6:03DB	JNP 04BD	, 出错则转04BDH
0FE6:03DE	CALL 02BF	, 复原4个字节
0FE6:03E1	LEA SI,[0150]	, 缓冲区首址送SI
0FE6:03E5	CMP WORD PTR [SI],+00	, 缓冲区第一字为0吗?
0FE6:03E8	JNZ 041E	, 不为0则转041EH
0FE6:03EA	CMP WORD PTR [SI+02],+00	, 缓冲区第二字为0吗?
0FE6:03EE	JNZ 041E	, 不为0则转041EH
0FE6:03F0	MOV AH,3F	, 调用号送AH
0FE6:03F2	MOV CX,0004	, 字节数4送CX
0FE6:03F5	LEA DX,[0150]	, 缓冲区首址送DX
0FE6:03F9	INT 21	, 读文件的4字节到缓冲
		, 区
0FE6:03FB	JNB 0400	, 不出错则转0400H
0FE5:03FD	JMP 04BD	, 出错则转04BDH
0FE6:0400	CALL 02BF	, 复原4字节
0FE6:0403	LEA SI,[0150]	, 缓冲区首址送SI
0FE6:0407	CMP WORD PTR [SI],+00	, 缓冲区第一字为0吗?
0FE6:040A	JNZ 0412	, 不为0则转0412H
0FE6:040C	CMP WORD PTR [SI+02],+00	, 缓冲区第二字为0吗?

0FE6:0410	JZ	0430	, 为0则转0403H
0FE6:0412	MOV	AX, [014E]	, 空出区地址送AX
0FE6:0415	MOV	ES, AX	, AX值送ES
0FE6:0417	ES:		
0FE6:0418	ADD	[0000], AX	, 把AX值加入0000H字单元
0FE6:041C	JMP	03DE	, 转03DEH
0FE6:041E	MOV	DI, [SI]	, 装入字地址偏移量送DI
0FE6:0420	MOV	AX, [014E]	, 定位段值送AX
0FE6:0423	ADD	AX, [SI+02]	, AX加上装入字地址段值
0FE6:0426	MOV	ES, AX	, AX值送ES
0FE6:0428	MOV	AX, [014E]	, 定位段值送AX
0FE6:042B	ES:		
0FE6:042C	ADD	[DI], AX	, 定位段值加入重定位项
0FE6:042E	JMP	03CC	, 转03CCH
0FE6:0430	MOV	AH, 3E	, 调用号送AH
0FE6:0432	MOV	BX, [0154]	, 文件柄送BX
0FE6:0436	INT	21	, 关闭CCCC.OVR文件
0FE6:0438	LEA	SI, [0156]	, 0156H送SI
0FE6:043C	MOV	AX, [SI+0E]	, 取栈段值到AX
0FE6:043F	ADD	AX, [014E]	, AX加上定位段值
0FE6:0443	MOV	SS, AX	, AX值送SS
0FE6:0445	MOV	SP, [SI+10]	, 栈指针值送SP
0FE6:0448	MOV	AX, [SI+16]	, 第一个重定位模块的段
			, 值送AX
0FE6:044B	ADD	AX, [014E]	, AX加上定位段值
0FE6:044F	PUSH	AX	, AX值入栈
0FE6:0450	MOV	BX, [SI+14]	, 第一个重定位模块的偏
			, 移量送BX
0FE6:0453	PUSH	BX	, BX值入栈
0FE6:0454	SUB	AX, 0010	, AX值减去0010H
0FE6:0457	MOV	DS, AX	, AX值送DS
0FE6:0459	MOV	ES, AX	, AX值送ES
0FE6:045B	RETF		, 转CCCC.OVR入口执行
0FE5:045C	MOV	AX, 4200	, 调用号和方式号送AX
0FE6:045F	XOR	CX, CX	, CX清0
0FE6:0461	XOR	DX, DX	, DX清0
0FE6:0463	INT	21	, 把CCCC.OVR指针移到文
			, 件首
0FE6:0465	JNB	046A	, 不出错则转046AH

0FE6 : 0467	JMP	04BD	, 出错则转04BDH
0FE6 : 0469	NOP		
0FE6 : 046A	MOV	DX, [014E]	, 空出区首址送DX
0FE6 : 046E	XOR	SI, SI	, SI清0
0FE6 : 0470	MOV	AH, 3F	, 调用号送AH
0FE6 : 0472	MOV	CX, 8000	, 字节数8000H送CX
0FE6 : 0475	MOV	DS, DX	, DX值送DS
0FE6 : 0477	XOR	DX, DX	, DX清0
0FE6 : 0479	INT	21	, 读CCCC.OVR的8000H个
			, 字节到空出区
0FE6 : 047B	JNB	0480	, 不出错则转0480H
0FE6 : 047D	JMP	04BD	, 出错则转04BDH
0FE6 : 047F	NOP		
0FE6 : 0480	ADD	SI, AX	, SI加上本次实际读入字
			, 节数
0FE6 : 0482	CMP	AX, 8000	, 本次实际读入8000H字
			, 节吗?
0FE6 : 0485	JB	048F	, 小于则转048FH
0FE6 : 0487	MOV	DX, DS	, DS值送DX
0FE6 : 0489	ADD	CX, 0800	, DX加上0800H
0FE6 : 048D	JMP	0470	, 转0470H
0FE6 : 048F	CS :		
0FE6 : 0490	MOV	DS, [014E]	, 定位段值送DS
0FE6 : 0494	PUSH	DS	, DS值入栈
0FE6 : 0495	POP	ES	, DS值送ES
0FE6 : 0496	MOV	DX, SI	, 读入总字节数送CX
0FE6 : 0498	CALL	02EA	, 复原字符值
0FE6 : 049B	PUSH	CS	, CS值入栈
0FE6 : 049C	POP	DS	, CS值送DS
0FE6 : 049D	MOV	AX, [014E]	, 定位段值送AX
0FE6 : 04A0	SUB	AX, 0010	, AX值减去0010H
0FE6 : 04A3	MOV	DS, AX	, AX值送DS
0FE6 : 04A5	MOV	ES, AX	, AX值送ES
0FE6 : 04A7	MOV	SS, AX	, AX值送SS
0FE6 : 04A9	XOR	SP, SP	, SP清0
0FE6 : 04AB	PUSH	AX	, AX值入栈
0FE6 : 04AC	MOV	AX, 0100	, AX值置为0100H
0FE6 : 04AF	PUSH	AX	, AX值入栈
0FE6 : 04B0	RETF		, 转CCCC.OVR入口点执行


```

; *****
; *
; *          本子程序未用
; *
; *****

```

```

;
0FE6:04B1 ES:
0FE6:04B2 MOV DS,[DI+02]
0FE6:04B5 MOV CX,0004
0FE6:04B8 CTI
0FE6:04B9 REPZ
0FE6:04BA MOVSB
0FE6:04BB SLI
0FE6:04BC RET

```

```

;
0FE6:04BD PUSH CS          ; CS值入栈
0FE6:04BE POP ES          ; CS值送ES
0FE6:04BF MOV CX,FFFF     ; 字节计数FFFFH送CX
0FE6:04C2 XOR DI,DI       ; DI清0
0FE6:04C4 STOSB           ; AL值存入DI指向处
0FE6:04C5 MOV AH,4C       ; 调用号送AH
0FE6:04C7 INT 21          ; 返回DOS

```