# 启动多种DOS版本的系统软盘

于杨丽　洪朝煜　厦门大学物理系(361003)

微型计算机一问世，支持它的核心软件也应运而生。从1981年DOS 1.0开始，直到1987年DOS 3.30的推出，标志着磁盘操作系统日趋成熟。在这期间，我国PC机也开始得到迅速普及和发展。随之而来的各种中文操作系统、中文文字处理及数据库等汉化软件，纷纷涌现。但DOS的发展并未划上休止符，尔后几年间，DOS 4、DOS 5、DOS 6接二连三地出台。电脑用户们很快发现：不少中文应用软件却无法在DOS新版本上正常运行。当然，这不是应用软件开发者的过失，因为任何应用软件的开发者只能依赖现行DOS提供的环境平台研制出自己的产品。DOS的每次革新是值得称道的，但这种创造性与依赖性的矛盾，是应用软件与新版本DOS不兼容的根源之一。

笔者正是在频繁地更换操作系统的过程中突发奇想：能否将多种版本的DOS集于一盘中？经过对多种版本DOS启动过程的剖析，终于成功地把DOS 3.30或(DOS 4)、DOS 5和DOS 6的启动系统汇集于一张1.2M软盘上，按需要选择启动对应的操作系统。虽然此举并非根本上解决DOS与应用软件之间不兼容的矛盾，但对苦于无门的广大用户来说，不乏为自我解脱之招。现介绍其原理和制作方法，从中也可对软盘操作系统的启动有更多的了解。

## 一、原理

MS-DOS由三个系统文件组成，即IO.SYS、MSDOS.SYS、COMMAND.COM(在PC-DOS中为IBMBIO.COM、IBMDOS.COM、COMMAND.COM)，这三个文件分别管理微机的输入输出设备、操作系统核心功能和命令处理。不同版本的DOS，其系统文件的内容不同，版本越高管理功能越强。利用某种版本启动微机，该种版本的管理功能将控制整个微机系统，但无论哪种版本的DOS都需要将自己的三个系统文件载入微机系统的过程，这就是系统自举(BOOTING)。对软盘而言，自举是从位于0面0道第一个扇区的引导记录开始的。显然，要选择性启动DOS，首先必须了解引导记录的特点以及不同DOS版本引导记录的异同。下面以DOS 3.30、DOS 4、DOS 5和DOS 6为例，对跟制作有关的方面进行比较分析。

(1)它们有着相同的结构，即JMP跳转指令、OEM标志、磁盘I/O参数表、引导程序代码。(其中DOS 6一字不差地沿用了DOS 5的引导记录)。此外，它们的引导程序有着相同的引导步骤，即首先进行初始化→计算根目录表和磁盘数据区的起始绝对扇号→将根目录表的第一个扇区读入0:500H开始地址处→判断有无IO.SYS和MSDOS.SYS两文件→若有，则把IO.SYS调入内存→将控制权转予IO.SYS程序由它继续完成自举。各版本虽然有以上共同点，但并不意味着它们的引导记录可互换代用，譬如DOS 4的引导记录不能用于启动DOS 5的系统文件自举，反之亦然。其原因是引导程序运行时，要从I/O参数表取得数据并在某存储单元和寄存器中建立某些参数，以供IO.SYS程序接着使用。而不同版本的引导记录其I/O参数表、存储单元地址及所建立在寄存器的参数略有差异，这势必影响非对应版本的IO.SYS文件的正常调入和运行。值得一提的是DOS 5的引导记录可用于DOS 3.30系统文件的自举，反之则不然。这原因亦如同上所致。

从以上的分析知道，启动多种DOS系统文件需要利用各自对应的引导记录，但存放引导记录的位置只有一个，即0面0道第一扇区。这矛盾如何解决？研究过电脑病毒的人一定晓得：在操作系统型病毒中，引导记录扇区往往被病毒程序占据，原DOS的引导记录被移到别处，待病毒程序执行后才将引导记录从存放处读入内存，并开始自举。我们完全可以利用这种技巧(病毒程序中不乏令人赞叹的技巧，只可惜用于邪门)，先将各版的DOS引导记录放置在某个连续扇区中，选择DOS版本的主程序放入0面0道第一个扇区，到时即可择需调入对应的引导记录了。

(2)它们的引导程序在调入IO.SYS文件的具体做法不同。DOS 5以前的版本不单要求IO.SYS、MSDOS.SYS两文件存放在物理上连续的扇区中，而且还严格要求IO.SYS文件的首扇区也是磁盘文件数据区的首扇区(对于1.2M软盘，逻辑扇区号为1DH，首簇号为2)。DOS 5却只要求系统两文件的扇区物理上连续就行了，首簇号可以是任意的。我们庆幸DOS设计者作这一变动，这使我们可以利用"借躯还魂"之法给予解决。请仔细查看表(一)的根目录表，第1、2、3项为DOS 3.30的系统文件，第4、5、6项为DOS 5的系统文件(已改名)，第7、8、9项为DOS 6的系统文件(已改名)。DOS 3.30的IO.SYS文件的首簇号为2。待我们欲启动DOS 5时，只要将根目录表中IO.SYS、MSDOS.SYS和COMMAND.COM文件的首簇号和长度全部改成DOS 5对应的系统文件(即I50、M50、C50)的首簇号和长度即可。这样DOS 5的引导程序在检查根目录表时仍可找到两个系统文件，在调入IO.SYS文件时恰恰引用了更改过的首簇号，正确地把I50文件调入内存，由于I50、M50存放时其扇区物理上是连

续的，所以自举全过程也同样正确地把 M50 调入内存。对于 COMMAND.COM 文件，此时 DOS 的核心功能已建立，只要首簇号无误，对应的 C50 文件也被调入执行。选择 DOS 6 的情况也一样。如果再选择 DOS 3.30，则把 DOS 3.30 系统文件的首簇号、长度恢复原样就行了。我们把 DOS 3.30 三个系统文件的首簇号和长度存放在主程序中 7DF2～7DFD。

**二、制作：**

以制作 MS-DOS 3.30、DOS 5 和 DOS 6 集于一盘为例。假设驱动器 A 为 1.2M 软盘驱动器。

(1)先在任意版本 DOS 系统状态下格式化一张 1.2M 软盘 FORMAT A:/S(笔者是在 DOS 3.30 状态)。按表(一)中目录项之排列依次将 DOS 5 和 DOS 6 的三个系统文件更名复制到这新盘上(DOS 5 的 IO.SYS 改名为 I50，MSDOS 改名为 M5.0，COMAND.COM 改名为 C50，依此类推。)有两个原则要记住，其一是 DOS 3.30 和 DOS 4 只能选择一个；其二，它的系统文件排列时应放在根目录表的前项，IO.SYS 文件的首簇号只能为 2。为了保证各系统文件 IO.SYS 和 MSDOS.SYS 所存放的扇区在物理上是连续的，应在盘上无其它非系统文件时，按顺序逐一用 COPY 命令复制。读者可选用其它方法复制，但要保证 IO.SYS 和 MSDOS.SYS 簇链的簇号是连续的。制作上失败，问题往往出在这里。

(2)用 DEBUG 将任意版本的软盘引导记录调入内存地址 7C00 处(笔者用 DOS 5 的引导记录)，即打入 L7C00 1 0 0，从 7C00 处开始输入主程序。若用 DOS 5 或 DOS 6 的引导记录，7C00 处的跳转指令 JMP 7C3E 不用更改，并且保留 7C02～7C3D 的 OEM 标志和 I/O 参数表，事实上主程序从 7C3E 开始输入。若用其它引导记录，虽然其 I/O 参数表较短，但为了主程序的正常运行，仍须将 7C00 处改成 JMP 7C3E，也要从 7C3E 处开始输入主程序。我们编写主程序时，设计最多可选择 4 种版本的 DOS。主程序地址 7D70 处"33"为版本数量的 ASCII 码，表示有三种版本。若有四种版本，则改成"34"。在 7D98～7DBD 中，按根目录的顺序输入对应的版本名。在此例中，第 4 个不使用，故填入"(4)DOS 0.0"或空着，这些只用于屏幕显示。另外，在 7DF2～7DFD 处的数据中，02、00、7E、57 为 DOS 3.30 的 IO.SYS 文件之首簇号和长度，2E、00、B0、75 和 69、00、DC、62 分别是 MSDOS.SYS 和 COMMAND.COM 的。主程序输入完毕，将新盘插入 A 驱动器中，键入 W7C00 0 0 1↙，将主程序存到新盘的引导记录扇区中。

(3)最后把 DOS 3.30、DOS 5 和 DOS 6 的引导记录存入新盘中。我们选择根目录表的最后 4 个扇区，即逻辑扇区的 19、1A、1B、1C 号作为这些引导记录的存放位置。当然，根目录表的目录项将减少 32 个，但这对于目录项容量为 224 个的 1.2M 软盘来说影响不大。按主程序的定位第一个 DOS 引导记录存入 19 号逻辑扇，第 2 个 DOS 引导记录存入 1A 号逻辑扇，依次类推。利用 DEBUG 时，可事先将 DOS 3.30 盘放进驱动器 A 中，键入 L100 0 0 1↙接着换上新盘并键入 W100 0 19 1↙，如此下去直到全部引导记录存完。在此例中 1C 号扇空着。

(4)可把除了 COMMAND.COM 以外的所有系统文件的属性改成隐含系统文件，使之隐藏起来。建立三个子目录取名为 DOS3.3、DOS50 和 DOS60，择优把对应版本的外部命令文件复制到相应的子目录中。这样可启动多种版本的系统盘之制作宣告完成。

**表一**

```
-d500 63f
1162:0500  49 4F 20 20 20 20 20 20-53 59 53 20 00 00 00 00   IO      SYS ....
1162:0510  01 00 00 00 00 00 01 00-42 10 02 00 7E 57 00 00   ........B...~W..
1162:0520  40 53 44 4F 53 20 20 20-53 59 53 20 00 00 00 00   @SDOS   SYS ....
1162:0530  00 00 00 00 00 00 00 01-00-42 10 2E 00 00 75 00 00   .........B....u..
1162:0540  43 4F 4D 4D 41 4E 44 20-43 4F 4D 20 00 00 00 00   COMMAND COM ....
1162:0550  00 00 00 00 00 00 01 00-42 10 68 00 DC 62 00 00   ........B.h..b..
1162:0560  49 35 30 20 20 20 20 20-20 20 20 20 00 00 00 00   I50             ....
1162:0570  00 00 00 00 00 00 01 00-28 6B 17 CF 00 96 82 00 00   ........(k......
1162:0580  4D 35 30 20 20 20 20 20-20 20 20 20 00 00 00 00   M50             ....
1162:0590  00 00 00 00 00 00 01 00-28 6B 17 11 01 12 92 00 00   ........(k......
1162:05A0  43 35 30 20 20 20 20 20-20 20 20 20 00 00 00 00   C50             ....
1162:05B0  00 00 00 00 00 00 01 00-28 6B 17 5B 01 E5 BA 00 00   ........(k.[....
1162:05C0  49 38 30 20 20 20 20 20-20 20 20 20 00 00 00 00   I80             ....
1162:05D0  00 00 00 00 00 00 00 00-30 6A 1A B9 01 16 9E 00 00   ........0j......
1162:05E0  4D 38 30 20 20 20 20 20-20 20 20 20 00 00 00 00   M80             ....
1162:05F0  00 00 00 00 00 00 00 00-30 6A 1A 09 02 FA 94 00 00   ........0j......
1162:0600  43 38 30 20 20 20 20 20-20 20 20 20 00 00 80 00   C80             ....
1162:0610  00 00 00 00 00 00 00 00-30 6A 1A 64 02 BD CE 00 00   ........0j.d....
1162:0620  41 42 4C 53 50 41 43 45-42 49 4E 20 00 00 00 00   DBLSPACEBIN ....
1162:0630  00 00 00 00 00 00 00 00-30 6A 1A BC 02 0E C8 00 00   ........0j......
```

另外，当把 PC-DOS 与 MS-DOS 混合于一盘时(如 PC-DOS 3.31，MS-DOS 5，MS-DOS 6 集于一盘)，除了把 PC-DOS 3.31 的二个系统文件 IBMBIO.COM 和 IBMDOS.COM 改成 IO.SYS 和 MSDOS.SYS 外，PC-DOS 3.31 引导记录中用于核查系统文件的存在的系统文件名标志，也要修改成 IO.SYS 和 MSDOS.SYS，才存入盘中 19 号逻辑扇。

使用时，将这"万用"启动盘插入 A 驱动器，冷或热启动后屏幕会显示当前的版本号 VERSION:3.30，并提示击任意键或选择版本。若击任意键，主程序则启动当前版本的操作系统；若要启动其它版本 DOS 可键入对应序号。由于在改变版本时主程序需改写根目录中系统文件的首簇号和长度，所以此时软盘不应贴有写保护纸纤，下次用此盘启动时，当前版本状态将是上一次选择的版本号。版本号的标志存于根目录表上，参看表(一)中 510H 处，此标志是主程序自动写入，用户无需干预。

下面介绍主程序中几个主要功能：

①7C5B～7C67：读根目录表第一个扇区。

②7CA3～7CBB：恢复第一个 DOS 系统文件的首簇号和长度。

③7CC0～7CE4：修改第一个 DOS 系统文件的首簇号和长度。

④7CE7～7CF3：改写根目录表。

⑤7CF7～7D0A：将所选择的 DOS 引导记录读入内存 7C00 处。

⑥7D28：将控制权转予引导记录程序。

```
14DE:7C00  EB3C            JMP     7C3E
14DE:7C00 90 4D 53 44 4F 53-35 2E 30 00 02 01 01 00
```

```
14DE:7C10 02 E0 00 60 09 F9 07 00-0F 00 02 00 00 00 00 00
14DE:7C20 00 00 00 00 00 00 29 FF-0F 33 1F 4E 4F 20 4E 41
14DE:7C30 4D 45 20 20 20 20 46 41-54 31 32 20 20 20
14DE:7C3E FA            CLI
14DE:7C3F 31C0          XOR AX,AX
14DE:7C41 8ED0          MOV SS,AX
14DE:7C43 8EC0          MOV ES,AX
14DE:7C45 8ED8          MOV DS,AX
14DE:7C47 D8007C        MOV AX,7C00
14DE:7C4A 89C4          MOV SP,AX
14DE:7C4C FB            STI
14DE:7C4D 89C6          MOV SI,AX
14DE:7C4F BF007E        MOV DI,7E00
14DE:7C52 FC            CLD
14DE:7C53 B90001        MOV CX,0100
14DE:7C56 F3            REPZ
14DE:7C57 A5            MOVSW
14DE:7C58 E90002        JMP 7E5B
14DE:7C5B BB0005        MOV BX,0500
14DE:7C5E B80102        MOV AX,0201
14DE:7C61 BA0001        MOV DX,0100
14DE:7C64 B90100        MOV CX,0001
14DE:7C67 CD13          INT 13
14DE:7C69 A11005        MOV AX,[0510]
14DE:7C6C 08C0          OR AL,AL
14DE:7C6E 7504          JNZ 7C74
14DE:7C70 40            INC AX
14DE:7C71 A31005        MOV [0510],AX
14DE:7C74 BE727F        MOV SI,7F72
14DE:7C77 E8B300        CALL 7D2D
14DE:7C7A E8BF00        CALL 7D3C
14DE:7C7D BE7C7F        MOV SI,7F7C
14DE:7C80 E8AA00        CALL 7D2D
14DE:7C83 31C0          XOR AX,AX
14DE:7C85 CD16          INT 16
14DE:7C87 30E4          XOR AH,AH
14DE:7C89 3C31          CMP AL,31
14DE:7C8B 7C6A          JL 7CF7
14DE:7C8D 3B06707F      CMP AX,[7F70]
14DE:7C91 7764          JA 7CF7
14DE:7C93 2D3000        SUB AX,0030
14DE:7C96 3B061005      CMP AX,[0510]
14DE:7C9A 745B          JZ 7CF7
14DE:7C9C A31005        MOV [0510],AX
14DE:7C9F 3C01          CMP AL,01
14DE:7CA1 751D          JNZ 7CC0
14DE:7CA3 BEF27F        MOV SI,7FF2
14DE:7CA6 BF1A05        MOV DI,051A
14DE:7CA9 E8AC00        CALL 7D58
14DE:7CAC BEF67F        MOV SI,7FF6
14DE:7CAF BF3A05        MOV DI,053A
14DE:7CB2 E8A300        CALL 7D58
14DE:7CB5 BEFA7F        MOV SI,7FFA
14DE:7CB8 BF5A05        MOV DI,055A
14DE:7CBB E89A00        CALL 7D58
14DE:7DBE EB27          JMP 7CE7
14DE:7CC0 48            DEC AX
14DE:7CC1 B96000        MOV CX,0060
14DE:7CC4 F7E1          MUL CX
14DE:7CC6 051A05        ADD AX,051A
14DE:7CC9 50            PUSH AX
14DE:7CCA 5E            POP SI
14DE:7CCB 56            PUSH SI
14DE:7CCC BF1A05        MOV DI,051A
14DE:7CCF E88600        CALL 7D58
14DE:7CD2 5E            POP SI
14DE:7CD3 83C620        ADD SI,+20
14DE:7CD6 56            PUSH SI
14DE:7CD7 BF3A05        MOV DI,053A
14DE:7CDA E87B00        CALL 7D58
14DE:7CDD 5E            POP SI
14DE:7CDE 83C620        ADD SI,+20
14DE:7CE1 BF5A05        MOV DI,055A
14DE:7CE4 E87100        CALL 7D58
14DE:7CE7 BB0005        MOV BX,0500
14DE:7CEA B80103        MOV AX,0301
14DE:7CED BA0001        MOV DX,0100
14DE:7CF0 B90100        MOV CX,0001
14DE:7CF3 CD13          INT 13
14DE:7CF5 7220          JB 7D17
14DE:7CF7 BD0500        MOV BP,0005
14DE:7CFA BB007C        MOV BX,7C00
14DE:7CFD 8B0E1005      MOV CX,[0510]
14DE:7D01 83C10A        ADD CX,+0A
14DE:7D04 B80102        MOV AX,0201
14DE:7D07 BA0001        MOV DX,0100
14DE:7D0A CD13          INT 13
14DE:7D0C 7311          JNB 7D1F
14DE:7D0E 29C0          SUB AX,AX
14DE:7D10 CD13          INT 13
14DE:7D12 4D            DEC BP
14DE:7D13 7402          JZ 7D17
14DE:7D15 EBE6          JMP 7CFD
14DE:7D17 BEC17F        MOV SI,7FC1
14DE:7D1A E81000        CALL 7D2D
14DE:7D1D EBFE          JMP 7D1D
14DE:7D1F E81A00        CALL 7D3C
14DE:7D22 BEBE7F        MOV SI,7FBE
14DE:7D25 E80500        CALL 7D2D
14DE:7D28 EA007C0000    JMP 0000:7C00
14DE:7D2D AC            LODSB
14DE:7D2E 08C0          OR AL,AL
14DE:7D30 7409          JZ 7D3B
14DE:7D32 B40E          MOV AH,0E
14DE:7D34 BB0700        MOV BX,007
14DE:7D37 CD10          INT 10
14DE:7D39 EBF2          JMP 7D2D
14DE:7D3B C3            RET
14DE:7D3C A11005        MOV AX,[0510]
14DE:7D3F 48            DEC AX
14DE:7D40 B90A00        MOV CX,000A
14DE:7D43 F7E1          MUL CX
```

```
14DE:7D45 059A7F        ADD AX,7F9A
14DE:7D48 89C6          MOV SI,AX
14DE:7D4A B90700        MOV CX,0007
14DE:7D4D AC            LODSB
14DE:7D4E B40E          MOV AH,0E
14DE:7D50 BB0700        MOV BX,0007
14DE:7D53 CD10          INT 10
14DE:7D55 E2F6          LOOP 7D4D
14DE:7D57 C3            RET
14DE:7D58 B90400        MOV CX,0004
14DE:7D5B AC            LODSB
14DE:7D5C AA            STOSB
14DE:7D5D E2FC          LOOP 7D5B
14DE:7D5F C3            RET
14DE:7D60 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
14DE:7D70 33 00 56 45 52 53 49 4F-4E 20 3A 00 0D 0A 73 74
14DE:7D80 72 69 6B 65 20 61 6E 79-6B 65 79 20 6F 72 20 53
14DE:7D90 45 4C 45 43 54 3A 20 28-31 29 64 6F 73 33 2E 33
14DE:7DA0 20 28 32 29 64 6F 73 35-2E 30 20 28 33 29 64 6F
14DE:7DB0 73 36 2E 30 20 28 34 29-64 6F 73 30 2E 30 0D 0A
14DE:7DC0 00 5B 57 52 49 54 45 20-50 52 4F 54 45 43 54 20
14DE:7DD0 6F 72 20 64 69 73 6B 20-65 72 72 6F 72 5D 00 00
14DE:7DE0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
14DE:7DF0 00 00 02 00 7E 57 2E 00-B0 75 69 00 DC 62 00 00
```

# 用 BASIC 程序读取 WPS 文件内容

丁怀德　山东聊城商业技工学校(252000)

WPS 的 D 编辑方式可以对文件进行加密，其密码最多不超过 8 个 ASCII 字符(除回车符外)。方法是把每个字符的 ASCII 码求反，然后再进行高低位互换，这八个字节从偏移地址 02DDH 开始存放，没设置密码时全为 00H，先设置后又取消，第一个字节变为 00H，而后面的内容却还是原来的值。改变密码，若新密码比旧密码短，旧密码多出的字节仍保留，这部分与新密码的最后一字节用 00H 隔开。

WPS 文件的文件头长度为 1024 字节。第 1，2 字节存放标志：01FF 或 02FF 或 03FF。正文的每个字符(包括各种控制字符)从偏移地址 0400H 处开始存放。如果没有设置密码或设置密码后又取消，存放的是每个字的机内码，否则，就用每 8 个字节与 02DDH 开始的 8 个字节进行异或运算，以其结果取而代之，直到遇到结束符为止。

根据这个方法，我用 BASIC 语言编写了一个程序(RDWPS.BAS)，它具有以下功能：

1. 可以读取并显示文本文件内容。

2. 可以破译以 D 方式编辑存盘的 WPS 文件密码。

3. 显示以 D 方式编辑存盘的 WPS 文件内容，无论是否加密。(不能显示 WPS 所特有的控制符号)。

4. 可以在只有低密软驱的 PC 机上完成以上功能。

加密后的文件比未加密时的显示速度要慢一些。如果需要，可以打印出来，方法是在打入命令 RUN 之前先按下 Ctrl+P 键。

在装有 WPS 系统的微机中，如果文件加密后忘记了密码，你只要按本程序所显示的密码，就能打开加密文件。

注：这个程序适用于 Super CCDOS 5.1，WPS 2.1。

```
30 DIM TB$(20),K(20)
40 PRINT:INPUT"请输入文件名";WJM$
50 OPEN WJM$ AS#1 LEN=2
60 FIELD #1,1 AS MA$,1 AS MB$
70 GET #1,1:TP$=MA$:TQ$=MB$
90 XP$=HEX$(ASC(TP$)):XQ$=HEX$(ASC(TQ$)):CLOSE
100 IF XP$+XQ$="1FF" THEN GOTO 120
110 BZ=1:GOTO 430
120 PRINT"以下为";WJM$;"的密码:":PRINT
130 OPEN WJM$ AS#1 LEN=1:FIELD #1,1 AS MA$
150 FOR K=734 TO 741
170 GET #1,K:W$=MA$:X$=HEX$(ASC(W$))
190 IF LEN(X$)=1 THEN X$="0"+X$
200 MM$=MM$+X$
210 IF X$="0"THEN 360
220 X$=RIGHT$(X$,1)+LEFT$(X$,1):GOSUB 1020
230 A=255-S
240 IF A=0 THEN PRINT"功能键(F1—F12)":GOTO 350
250 IF A>32 AND A<127 THEN PRINT CHR$(A);" ";:GOTO 350
260 IF A=127 THEN PRINT "Ctrl+Backspace":GOTO 350
270 IF A<27 THEN PRINT "Ctrl+";CHR$(A+64):GOTO 350
280 ON A-26 GOTO 300,310,350,320,330,340
290 GOTO 350
300 PRINT "Esc":GOTO 350
310 PRINT"Ctrl+L":GOTO 350
320 PRINT"Ctrl+ ^ ":GOTO 350
330 PRINT "Ctrl+O":GOTO 350
340 PRINT "空格"
350 NEXT K
```