

# WPS最新版本(6.0F)源代码剖析

莫文 卓强 柯龙 编



电子工业出版社

责任编辑:杨悦秦  
封面设计:钟山

## 欢迎订阅我社计算机资料

尊敬的读者:你们好!

为适应计算机产业的开发应用,84年以来我社编辑出版了大批的计算机图书资料。由于选材新颖、内容丰富、实用性强,深受广大读者欢迎和有关专家的好评,销全国各省市,销售量每年大幅度上升。为不断改进编辑、出版、发行工作,殷切希望读者提出宝贵意见,一旦被采纳,我们将给予奖励。对确系书刊质量问题负责免费调换。从邮局或银行预汇款者,三天内我社将书挂号发出,收取10%的包装邮寄费。一次购书款500元以上可按批发价给予优惠。

欢迎投稿,欢迎经销!

银行汇至:西安市工商银行解放路办事处

户名:陕西电子杂志社

帐号:144221-44

电话:(029)79-4344;79-4391

电挂:8860

地址:西安市西五路16副5号甲

邮政编码:710004

陕西电子杂志社

陕西省图书准印证  
陕新出批字[92]第359号

# WPS 最新版本(6.0F)源代码剖析

莫文 卓强 柯龙 编

陕西电子杂志社

## 内容提要

WPS 自问世以来，因其优越的性能，良好的用户界面，领先的支持技术，以及迅速地升级换代，受到广大用户的欢迎，成为最广泛使用的字处理系统。其核心支撑软件是精心设计的汉字操作系统 spdos。spdos 集中体现了汉字操作系统的优点，并具有其它汉字系统所没有的特色，如：

- 可装拆的模块结构
- 外挂输入法通用接口
- 显示器类型自动识别
- 支持西文软件直接写屏，西文软件无需汉化可直接使用中文
- 自动识别中文字符和西文制表符
- 强大的功能菜单
- 简繁体一键转换
- 可直接运行 ORACLE 数据库等等。

了解和研究 spdos 的内部细节，对每一个与汉字操作系统打交道的人来说都大有裨益。为此，我们对 spdos 实现的全过程进行了分析和解释，并加以整理，愿与广大读者共勉。

本书共分为五章。第一章介绍 spdos 初始化过程。第二章是键盘管理模块。第三章是显示管理模块。第四章是打印管理模块。第五章是扩展内存管理模块。每一章都简要说明了各模块的功能和特色，然后给出源程序剖析。为了便于读者阅读时查找，程序中的所有标号都用加载后内存绝对地址的形式给出。

本书的主要读者是从事与汉字有关的计算机科研、设计和应用的工作人员，尤其是计算机系统设计人员。本书涉及了许多 DOS 编程技术和概念，所以它也是深入了解和熟练掌握 DOS 编程技术的难得的一本参考书。

由于作者水平有限，书中错误之处，恳请各位同行指正。

作 者

一九九三年五月

# 目 录

第一章 初始化模块 .....	( 1 )
1.1 初始化过程 .....	( 1 )
1.2 源程序分析 .....	( 2 )
第二章 键盘管理模块 .....	(30)
2.1 工作原理 .....	(30)
2.2 功能调用 .....	(31)
2.3 重要特性说明 .....	(33)
2.4 源程序分析 .....	(34)
第三章 显示管理模块 .....	(143)
3.1 工作原理 .....	(143)
3.2 功能调用 .....	(143)
3.3 重要特性说明 .....	(148)
3.4 源程序分析 .....	(148)
第四章 打印管理模块 .....	(242)
4.1 打印系统功能概述 .....	(242)
4.2 屏幕拷贝驱动程序(INT 05H) .....	(245)
4.3 打印驱动程序(INT 17H) .....	(247)
4.4 源程序分析 .....	(249)
第五章 扩充内存管理模块 .....	(304)
5.1 内存扩充技术 .....	(304)
5.2 SPDOS 扩充内存管理 .....	(307)
5.3 源程序分析 .....	(308)
附录 1 扩充输入法实例 .....	(318)
附录 2 汉字系统和打印机参数 .....	(324)

# 第一章 初始化模块

## 1.1 初始化过程

运行 spdos, 即进行系统初始化。

SPDOS 首先检查机器内是否配有 PUC Super 汉卡, 如果没有, 则报告 PUC Super 汉卡没有安装, 然后退出。否则, 就将字库读写程序驻留内存, 然后:

(1). 初始化显示模块:

识别显示器类型, 根据其类型设定显示器参数、初始化程序及设置颜色程序, 设置显示中断(INT 10H)的入口地址。

显示模块的核心是显示中断服务子程序。它通过软中断 INT 10H 调用来完成。

如果显示程序识别不出显示器的类型, 则报告“Video parameter not set!”, 表示用户所用的显示器系统不认识, 需要用户提供显示器类型参数。

(2). 初始化输入模块:

设置键盘中断 (INT 16H) 的入口地址。将显示器设置成图形方式, 显示版本号、公司版权和版本日期。

基本输入模块的核心是输入码处理和对用户的 I/O 接口。它本身包括英文数字输入法、国标区位输入法或多功能拼音输入法, 其它输入法则在扩充输入法里。

(3). 如果有的话, 读扩充词组文件到内存指定区域。

(4). 根据需要设置扩充内存使用中断

其初始化流程如图 1-1 所示

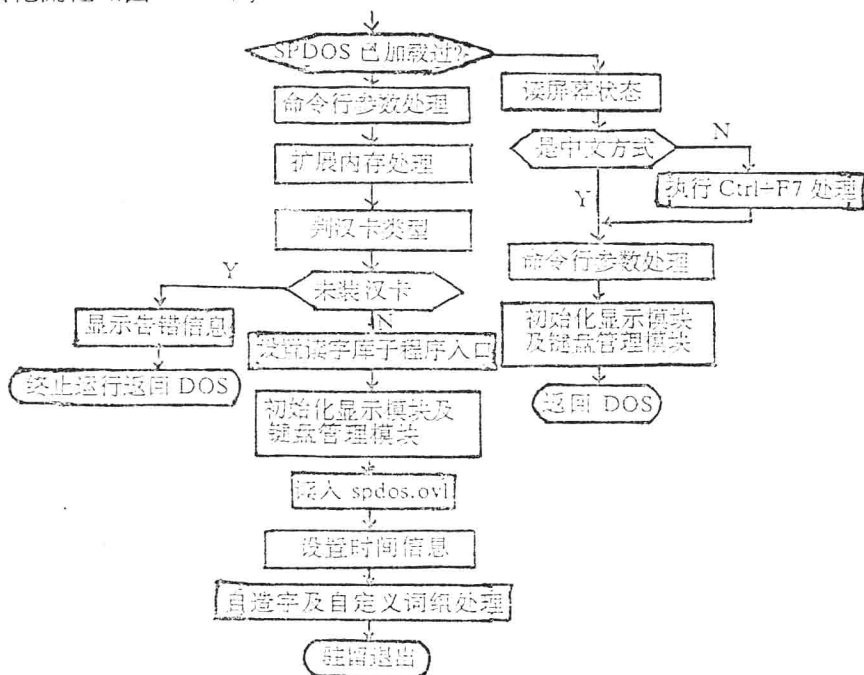


图 1-1 初始化流程

## 1.2 源程序分析

;(主程序入口在 L\_F90F 处)

;自选字及自定义词组处理

L\_F141:

```
MOV    AX, 0F368H    ; 取得数据段起始位移
MOV    CL, 4
SHR    AX, CL        ; 计算段值
MOV    CX, DS
ADD    AX, CX        ; 取得数据段段址
MOV    D_F32E, AX    ; 保存段值
CALL   S_F2B5        ; 读入自定义字数据
CMP    BYTE PTR D_F330, 0    ; 有扩充词组文件?
JE     L_F16A        ; 否, 转
MOV    DX, 0F330H
MOV    AX, 3D00H
INT    21H           ; 打开文件
JNC    L_F177        ; 成功, 转
MOV    DX, 0F350H
MOV    AH, 9
INT    21H           ; 显示出错信息“未找到词组文件”
```

;驻留退出

L\_F16A:

```
MOV    DX, D_F32E
MOV    AX, CS
SUB    DX, AX
MOV    AX, 3100H
INT    21H
```

扩充词组文件处理

L\_F177:

```
MOV    SP, 100H
MOV    BX, AX
MOV    ES, D_F32E    ; ES = 词组段的段值
XOR    BP, BP
XOR    DI, DI
```

L\_F184:

```
MOV    AH, 3FH
MOV    CX, 0AH
MOV    DX, 0F330H
INT    21H           ; 读 0A 个字节至 D_F330 处
CMP    AX, 8         ; 实际读出字节数小于 8?
JB     L_F203        ; 是, 转
TEST   BYTE PTR D_F333, 80H    ; 第 4 个字节为汉字(多字词组)?
JNZ    L_F20B        ; 是, 转
```

;处理自定义双字词组

```
MOV SI, DX ;SI 指向数据首部
```

;读入四个双拼输入码

```
LODSW
```

```
MOV DX, AX
```

```
LODSW
```

```
CMP AL, 61H ;a'
```

```
JNE L_F1A6
```

```
MOV AL, 76H ;v'
```

L\_F1A6:

```
CMP DL, 61H ;a'
```

```
JNE L_F1AD
```

```
MOV DL, 76H ;v'
```

;将四个输入码压缩为 20 位, 每个输入码占 5 位;

;将两个汉字压缩为 28 位, 每个汉字占 14 位;

;这样每个词组项在内存占 6 个字节(共 48 位)

L\_F1AD:

```
AND AX, 1F1FH
```

```
AND DX, 1F1FH
```

```
XCHG AX, DX
```

```
XCHG AL, AH
```

```
XCHG DL, DH
```

```
MOV CL, 3
```

```
SHL DL, CL
```

```
SHL DX, CL
```

```
SHL DX, 1
```

```
RCL AL, 1
```

```
SHL DX, 1
```

```
RCL AL, 1
```

```
SHLDX, 1
```

```
RCL AL, 1
```

```
SHL DX, 1
```

```
RCL AX, 1
```

```
SHL DX, 1
```

```
RCL AX, 1
```

```
SHL DX, 1
```

```
RCL AX, 1
```

```
STOSW
```

```
LODSW
```

```
MOV CX, AX
```

```
LODSW
```

```
XCHG CL, CH
```

```
XCHG AL, AH
```

```
SHL AL, 1
```



```

SHL    AX, 1
SHL    AX, 1
RCL    CL, 1
SHL    AX, 1
RCL    CX, 1
SHR    AL, 1
SHR    AL, 1
SHR    AL, 1
SHR    AL, 1
OR     AL, DH
STOSB
MOV    AL, CH
STOSB
MOV    AL, CL
STOSB
MOV    AL, AH
STOSB
INC    BP
JMP    SHORT L_F184

```

;读取的实际字节数小于8

L\_F203:

```

CALL    S_F2A2           ;设置词组区参数
XOR    AX, AX
STOSW           ;设置词组区结束标志
JMP    SHORT L_F267

```

;多字词组处理

L\_F20B:

```

PUSH    AX
CALL    S_F2A2           ;设置词组区参数
POP     BP
MOV    SI, 0F330H       ;指向缓冲区
ADD    BP, SI

```

L\_F215:

;读取三个输入码

```

CALL    S_F27F
JC     L_F264
MOV    DH, AL
CALL    S_F27F
JC     L_F264
MOV    DL, AL
CALL    S_F27F
JC     L_F264

```

;将这三个码压缩为15位,存放在两个字节中;

```

AND    DX, 1F1FH

```

```

    AND    AL, 1FH
    MOV    CL, 3
    SHL    AL, CL
    SHL    AL, 1
    RCL    DL, 1
    SHL    AL, 1
    RCL    DL, 1
    SHL    AL, 1
    RCL    DL, 1
    SHL    AL, 1
    RCL    DX, 1
    SHL    AL, 1
    RCL    DX, 1
    MOV    AX, DX
    XCHG   AH, AL
    STOSW
    MOV    DX, 2
;读入汉字码
L_F24E:
    CALL   S_F27F
    JC     L_F262
    CMP    AL, 0D1H
    JE     L_F25B
    STOSB
    INC    DX
    JMP    SHORT L_F24E
L_F25B:
    XOR    DX, DX
    CALL   S_F27F
    JNC    L_F215
L_F262:
    SUB    DI, DX
;存入词组文件结束符
L_F264:
    XOR    AX, AX
    STOSW
;计算驻留长度,并驻留退出
L_F267:
    MOV    AX, DI
    ADD    AX, 0FH
    MOV    CL, 4
    SHR    AX, CL
    MOV    BX, ES
    ADD    AX, BX

```

```

MOV    BX, DS
SUB    AX, BX
MOV    DX, AX
MOV    AX, 3100H
INT    21H
;=====
; SUBROUTINE
;=====
;子程序: 读文件
S_F27F PROC    NEAR
    CMP    SI, BP        ;SI 小于 BP?
    JB     L_F29F        ;是, 转
    PUSH  CX
    PUSH  DX
;读 38H 个字节至 F330H 处
    MOV    AH, 3FH        ;?
    MOV    DX, 0F330H
    MOV    CX, 38H
    NOP
    INT    21H
    POP    DX
    POP    CX
    MOV    SI, 0F330H
;调整指针
    MOV    BP, AX
    ADD    BP, SI
    AND    AX, AX        ;文件已结束?
    JNZ    L_F29F        ;否, 转
    STC                    ;置进位标志
    RETN
L_F29F:
    LODSB
    CLC                    ;清除进位标志
    RETN
S_F27F ENDP
;=====
; SUBROUTINE
;=====
;子程序: 设置词组区参数
S_F2A2 PROC    NEAR
    MOV    D_D74C, BP        ;保存自定义双字词组个数
    MOV    WORD PTR D_D74A, ES ;保存词组区段址
    MOV    AX, BP
    MOV    CX, 6

```

```

        MUL     CX
        MOV     D_D74E, AX           ;保存多字词组区起始地址
        RETN
S_F2A2 ENDP
;=====
        ; SUBROUTINE
;=====
子程序: 读入自定义字点阵数据
S_F2B5 PROC    NEAR
        MOV     DX, 0FBD6H           ;指向文件名 "spword.dot"
        CALL    S_FD10              ;打开该文件
        JNC     L_F2BE              ;成功, 转
        RETN
L_F2BE:
        MOV     BX, AX               ;保存文件句柄
        MOV     AX, D_F32E
        MOV     D_F32C, AX          ;字数据区段值送入 D_F32C 中
        MOV     AX, 8000H
        MOV     DS, AX
        MOV     AH, 3FH
        MOV     CX, 4B50H
        XOR     DX, DX
        INT     21H                 ;从文件读取 4B50H 字节数据, 送至
                                     8000: 0000 处
        MOV     ES, CS: D_F32C      ;ES = 字数据区段值
        MOV     SI, 0AH
        XOR     DI, DI
        MOV     CX, 787H
L_F2E1:
        CMP     WORD PTR [SI], 0    ;字数据已结束?
        JE      L_F312              ;是, 转
        CMP     WORD PTR [SI+2], 1010H ; 是字数据索引标志
        JNE     L_F312              ;否, 转
        MOVSW                          ;读入国标码
        PUSH    SI
        PUSH    CX
        PUSH    DS
        MOV     DX, [SI+4]           ;取得数据区的低位偏移
        MOV     CX, [SI+6]           ;取得该字数据区的高位偏移
        MOV     AX, 4200H
        INT     21H                 ;移动文件指针
        PUSH    ES
        POP     DS
        MOV     DX, DI

```

```

MOV     CX, 20H
MOV     AH, 3FH
INT     21H                ;将该汉字的点阵数据读入工作区，共 32 字节
ADD     DI, 20H
POP     DS
POP     CX
POP     SI
ADD     SI, 8              ;指向下一索引项
LOOP    L_F2E1            ;继续读取点阵数据
L_F312:
XOR     AX, AX
STOSW                      ;设置自定义字点阵数据区结束标记
MOV     AX, CS
MOV     DS, AX
MOV     ES, AX
MOV     AH, 3EH
INT     21H                ;关闭文件
ADD     DI, 0FH
MOV     CL, 4
SHR     DI, CL
ADD     D_F32E, DI        ;重计算自定义词组区的段值
CLC
RETN
S_F2B5 ENDP
D_F32C DW 0                ; 工作单元
D_F32E DW 0                ; 工作单元
D_F330 DB 0                ; 工作单元
        DB 0, 0
D_F333 DB 0
;=====
; SUBROUTINE
;=====
;子程序：初始化显示管理模块，键盘管理模块，及日时钟模块
S_F3FB PROC NEAR
; 取 INT 10H 中断向量并保存
MOV     AX, 3510H
INT     21H
MOV     WORD PTR D_8370, BX
MOV     WORD PTR D_8370+2, ES
MOV     D_8378, 3
CMP     BYTE PTR D_FB28, 0    ;是否已运行过 spdos?
JNE     L_F418                ;是，转
MOV     AX, CS
MOV     ES, AX                ;否，ES = CS

```

```

L_F418:
    PUSH    ES
    CALL    S_F4F5                ;测试显示卡类型
    POP     ES
    CMP     WORD PTR DS: D_FB29, 0 ; 是否在命令行设置了参数
    JE      L_F428                ;否, 转
    MOV     SI, DS: D_FB29

L_F428:
    MOV     CX, 15H
    MOV     DI, 8359H
    CLI
    MOV     DX, ES: D_835F
    REP     MOVSB                  ;将显示卡参数送入工作区
    MOV     AX, ES: D_8359        ;读取扫描线条数
    MOV     BL, 12H               ;每行为 18 线
    DIV     BL
    DEC     AL                    ;计算得到显示行数
    MOV     ES: D_836F, AL        ;保存显示行数
    CMP     BYTE PTR DS: D_FB28, 0 ;是否已运行过 spdos?
    JE      L_F471                ;否, 转
    MOV     ES: D_839F, 1
    MOV     AX, 3
    INT     10H                   ;调用原 INT 10H 的 3 号功能
    MOV     ES: D_839F, 0
; 设置有关参数
    MOV     ES: D_8378, 3
    MOV     ES: D_8376, 3
    MOV     ES: D_8377, 7
    STI
    JMP     SHORT L_F4C5

L_F471:
    STI
; 建立新的 INT 10H 中断向量
    MOV     AX, 2510H
    MOV     DX, 9BD1H
    INT     21H
; 读取 INT 8H 中断向量, 并保存
    MOV     AX, 3508H
    INT     21H
    MOV     DS: D_8399, BX
    MOV     DS: D_839B, ES
; 设置新的 INT 8H 中断向量
    MOV     AX, 2508H
    MOV     DX, 0B426H

```

```

INT      21H
; 填充段地址
MOV      SI, 0C5BEH
MOV      CX, 0AH
MOV      AX, CS
L_F497:
MOV      [SI], AX
ADD      SI, 4
LOOP     L_F497
; 读取原INT 16H 中断向量并保存
MOV      DS: D_EC9E, AX
MOV      AX, 3516H
INT      21H
MOV      DS: D_C191, BX
MOV      DS: D_C193, ES
MOV      DX, 0C64CH
; 建立新的INT 16H 中断向量
MOV      AX, 2516H
INT      21H
; 调用原INT 10H 的3号功能调用
MOV      BYTE PTR DS: D_839F, 1
MOV      AX, 3
INT      10H
MOV      BYTE PTR DS: D_839F, 0
L_F4C5:
MOV      AX, CS
MOV      DS, AX
MOV      ES, AX
MOV      AX, 6
INT      10H
; 设置为6号图形显示模式
MOV      D_C1A5, 9
; 设置为半角英文输入方式
MOV      AH, 24H
INT      16H
MOV      AX, 2404H
INT      10H
; 清提示行, 显示提示行状态
MOV      DX, 0F370H
MOV      AH, 9
INT      21H
; 显示版本信息
MOV      AX, 2406H
; 屏幕宽度改为40列
INT      10H
MOV      AH, 2
MOV      DX, 100H
INT      10H
; 置光标位置
INT      1

```

```

        INT      3
        RETN
S_F3FB ENDP
;=====
        ; SUBROUTINE
;=====
;子程序: 判定显卡类型, 并返回相应参数区的起始地址 SI
S_F4F5 PROC    NEAR
        MOV     D_836E, 0
        PUSH   DS
        MOV     AX, 0C000H
        MOV     DS, AX                ;DS = 0C000H
        XOR    BX, BX
        CMP    WORD PTR [BX], 0AA55H ;是视频扩展 ROM 起始标志?
        JE     L_F50C                ;是, 转
        POP    DS                    ;恢复 DS
        JMP    L_F618                ;否, 转
;VGA / EGA 卡判定
L_F50C:
        MOV     ES, AX
        POP    DS
        MOV     AX, 1A00H
        XOR    BX, BX                ;调用原 INT 10H 的 1AH 号功能
                                         调用读 VGA 配置
                                         ;信息, 若返回 AL = 1AH, 则表示有效

        PUSHF
        CALL   DWORD PTR DS: D_8370
        CMP    AL, 1AH                ;是否有效?
        JE     L_F520                ;是 VGA 卡, 转
        JMP    L_F5A6                ;否, 转
;各类 VGA 卡的判别
L_F520:
        MOV     SI, 0FB24H
        MOV     CX, 3
        MOV     DI, 7DH
        REPE   CMPSB                  ;ES: 7DH 处是否有标志"VGA"?
        JNZ    L_F536                ;否, 转
        MOV     BYTE PTR DS: D_F864, 58H ;是, 设置有关参数
        MOV     SI, 0F7E9H
        RETN
L_F536:
        MOV     SI, 0FB24H
        MOV     CX, 3
        MOV     DI, 0CH

```



```

REPE    CMPSB                ;ES: 0CH 处是否有标志"VGA"?
JNZ     L_F54C                ;否, 转
MOV     BYTE PTR DS: D_F864, 52H 是, 置有关参数
MOV     SI, 0F7E9H
RETN

L_F54C:
MOV     SI, 0FB24H
MOV     CX, 3
MOV     DI, 0C8H
REPE    CMPSB                ;ES: 0CH 处是否有标志"VGA"?
JNZ     L_F567                ;否, 转
MOV     BYTE PTR DS: D F864, 5BH ; 是, 置有关参数
MOV     BYTE PTR DS: D 836E, 1
MOV     SI, 0F7E9H
RETN

L_F567:
MOV     SI, 0FB24H
MOV     CX, 3
MOV     DI, 0C6H
REPE    CMPSB                ;ES: 0CH 处是否有标志"VGA"?
JNZ     L_F582                ;否, 转
MOV     BYTE PTR DS: D F864, 5BH ; 是, 设置有关参数
MOV     BYTE PTR DS: D 836E, 1
MOV     SI, 0F7E9H
RETN

L_F582:
MOV     SI, 0FB24H
MOV     CX, 3
MOV     DI, 0BAH
REPE    CMPSB                ;ES: 0BAH 处是否有标志"VGA"?
JNZ     L_F59D                ;否, 转
MOV     BYTE PTR DS: D F864, 5BH ; 设置有关参数
MOV     BYTE PTR DS: D 836E, 1
MOV     SI, 0F7E9H
RETN

L_F59D:
MOV     BYTE PTR DS: D F864 , 5BH ; 对于其余类型的VGA卡, 设置
                                ;缺省参数
MOV     SI, 0F7E9H
RETN

;EGA卡判别
L_F5A6:
MOV     SI, 0FB1EH
MOV     CX, 6

```