

# 通用文件显示 TYPEE 程序

云南省军区自动化站 李晓华

**摘要:** 笔者把目前非常流行的几种应用软件(各种语言编辑器、编辑排版软件、数据库系统等)所产生的文件,分为三种基本类型的文件(如 *WS*、*WPS*、*FOXbase* 或 *dBASE*)。分析了这几种文件类型的基本格式,说明 *DOS* 内部 *TYPE* 命令的不足。提出解决问题的办法。编写了能自动分辨出这几种文件类型、显示文件内容的程序。对提高计算机的使用效率起到一定作用。

*PC-DOS* 的内部 *TYPE* 命令,能显示文件的内容,当要查看文件内容时,用它比较方便。但 *TYPE* 命令存在有不足之处,那就是不能显示诸如用 *WPS* 编辑系统、数据库系统等所产生的存盘文件。当要查看文件内容时,就不得不进入相应的应用系统,给用户操作、使用计算机带来不便,特别是对不熟悉数据库语言的用户,在使用计算机查看库文件内容时存在有一定的困难。

为减少操作人员使用上不必要的麻烦、提高计算机的工作和使用效率,使计算机能进一步得到广泛应用,笔者根据用户要求,编制了一个通用文件显示 *TYPEE* 程序。供大家使用。

从目前操作、使用计算机情况看,用得较多的文件有以下三大类:

1. 标准的文本文件。诸如用 *WS*、*EDLIN*、各种语言编辑器等所产生的存盘文件、源程序等。这种文件全部都是用户数据,不存在有其它附加信息。因此用户数据都是从文件偏移 *00H* 处开始存放的。

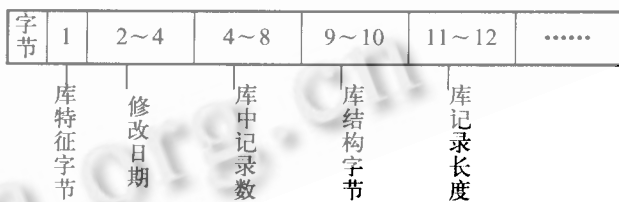
2. 非标准文本文件。诸如用 *WPS* 编辑系统等所产生的存盘文件。这种文件除用户数据外,还存在有其它本系统要用到的附加信息(如加密信息等),通常这些信息是放在用户数据前面。因此这种文件的用户数据不是从 *00H* 处开始存放的。

3. 数据库文件。诸如用数据 *dBASE*、*FOBASE* 等所产生的存盘库文件。这类库文件除用户数据外,还有其文件结构信息、结构字段、文件记录等。这类文件的存储格式稍微复杂些。但其用户数据都是以记录形式存放在文件中的。

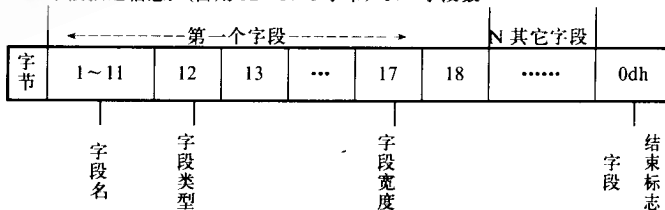
通过用 *DEBUG* 调试工具分析发现,三种文件的数据结构在机器内的存放格式不一样的。因而用 *TYPE* 的内部命令不能显示后两种类型的文件。经多次分析找到了三种文件类型数据结构的基本存放格式:

标准的文本文件是从文件偏移地址 *00H* 处开始存放用户数据的。

*WPS* 文件是从文件偏移地址 *100HH* 处开始存放用户数据的。在此之前是有关 *WPS* 系统内部的一些重要信息包括编辑过程中用到的系统参数、加密信息等。



字段描述信息: (占用  $32 * N + 1$  字节)  $N =$  字段数



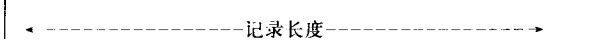
(field\_\_name) (field\_\_type) (field\_\_width)

数据库管理系统中的库文件,如 *FOXBASE* 的库文件是从库文件偏移地址  $32 + N * 32 + 1$  ( $N$  是数据中的字段数) 字节处开始存放数据的。其整个库结构由三部

分组成：库文件描述信息、字段描述信息、数据记录部分。

库文件描述信息：

数据描述信息：(占用 32 字节)



|   |   |          |     |
|---|---|----------|-----|
| 1 | * | 第一条记录数   |     |
| 2 | * | 第二条记录数   |     |
| N |   | 第 N 条记录数 | 1Ah |

\* 删除标志位

↑ 文件结束标志

只要知道了三种文件数据结构的基本存放格式后，就能很方便地编写读取三种类型文件内容的程序了。

本程序能自动分辨出三种文件类型，根据文件类型执行相应的过程。

该程序的使用方法非常简单，只要在 TYPEE 后输入其文件名即可。程序清单见下：

/\*

```

typee.c c 通用文件显示程序
使用方式: TYPEE 文件名.扩展名
程序说明: 1. TYPEE 主程序
            2. disp__txt() 显示标准的文本文件
            3. disp__nottxt() 显示非标准的文本文件
                (如WPS文件)
            4. disp__dm() 显示数据库文件
                (如FOXbase)
            5. 本程序稍加修改就能适用于其它编辑器所
                产生的文件。使之达到真正通用之目的。
编程语言: 采用 Turbo C 2.0 版本。
运行环境: 使用任何 PC-DOS 操作系统的计算机。
    
```

```

/ * * * * * TYPEE.c * * * * * /
#include <stdio.h>
#include <dir.h>
#include <string.h>
#include <dos.h>
void disp__txt(void);
    
```

```

void disp__nottxt(void);
void disp__dm(void);
FILE * fp;
char C;
unsigned char filename[11];
main (int arge, char * argv[])
{
    
```



```

puts(" ");
puts(" ");
puts(" * * * * * 欢迎你使用 * * * * * ");
puts(" * * * * * 通用文件显示程序 * * * * * ");
puts(" * * * * * 李晓华 版权所有 未经同意 不得复制 违者必究 * * * * * ");
puts(" ");
puts(" ");

if (argc > 1) strcpy (filename, argv[1]);
else{
printf("使用格式: TYPEE 文件名,扩展名");
exit(1);
}

if (fp = fopen(argv[1],"rb") == NULL){
printf("%s",filename);
printf("文件不存在或文件没有找到。\\");
exit(1);
}

c = getc(fp);
if(c == Ox01){
disp__nottxt();
exit(1);}
if = (c == Ox03){
disp__dm();
exit(1);}
if((c!= Ox01) (c!= Ox03)){
disp__txt();
exit(1);}
}
    
```

```

/ * * * * * disp__txt() * * * * *
* * /
void disp__txt (void)
{
puts("这是一个文本文件");
putchar('\\n');
}
    
```

```

fseed(fp,Ox00,0);
while(feof(fp) == NULL){
c = getc(fp);
putchar(c);
}
fclose(fp);
}
/ * * * * * disp__nottxt() * * * * *
* * /
void disp__nottxt(void)
{
fseek (fp,Ox2de,0);
if((c = getc(fp))!Ox0) {
puts ("这是一个 WPS 加密文件");
puts ("为保密信息,该文件不得显示!");
exit(1);}
else{
puts ("这是一个 WPX 文件");
putchar ('\n');
fseek(fp,Ox400,0);
while (feof(fp) == NULL){
c = getc(fp);
putchar(c);}
}
fclose(fp);
}
/ * * * * * * * * * * disp__dm() * * * * *
/
void disp__dm(void)
{
char buf[512],buf1[512];
char field__len,width[60], field__name[11],c;
int stru__len,rec__len,i1,i2;
unsigned date[3],total, i3,i4,i5;
unsigned char i6;
long rec__num, dat;
puts ("这是一个数据库文件。");
for (dat = 0;dat < 3;dat++) date[dat] = getc(fp);
fread(rec__num, 4,1,fp);
printf("\n 数据库结构:%s",filwname);
printf("\n 数据记录数:%Ld",rec__num);
printf("\n 最新更改日期:%2d: %2d: %2d",
date[0], date[1], date[2]);4total = getw(fp);
printf("\n * * 总和 * * :%d",total);
putchar ('\n');
fseek(fp,8L, SEEK__SET);/ * pointer stru byte
of number * /
fread (stru__len, sizeof(int),1,fp);
fseek (fp,32\1, SEEK__SET);/ * pointer field
first * /
for (i1 = 0;i1 < ((stru__len-(32+1) / 32);i1++){
fread (field__name, sizeof (char),10,fp);
field__name[11] = '\0';
fseek(fp,6L,SEEK__CUR);
fread(field__len,sizeof(char),1,fp);
width[i1] = field__len;
printf("%s",field__name);
for (i2 = 0,i2 < field__len;i2++)putchar('\n');
fseek(fp,15L,SEEK__CUR);
}
printf("\n");
fseek(fp,10L,SEEK__SET);
fread(rec__len,sizeof(int),1,fp);
fseek(fp,(long)stru__len+1,SEEK__SET);
fread(buf,rec__len,1,fp);
while(!feof(fp)){
for(i3 = i4 = i5 = 0;i3 < (stru__len-(32+1) / 32;i3++)
{
f o r
(i6 = 0;i6 < width[i3];i6++)buf1[i4++] = buf[i5++];
buf1[i4++] = '\x20';
buf1[i4++] = '\x20';
}4buf1[i4] = '\0';
printf("%s\n",buf1);
fread(buf,rec__len,1,fp);
}
fclose(fp);}

```