

自适应多种显示卡的屏幕保存 / 恢复程序

湖南省双峰工商银行 罗 辉

摘要: 本文提供的中西文屏幕信息的保存和恢复(SAVE/ RESTORE)程序,较好地解决了各种汉字系统下不同显示卡屏幕信息的保存和恢复,也是对 FOXBASE+的 SAVE/ RESTORE ACREEN 命令的补充。

曾有许多刊物刊登了一些用于保存和恢复汉字系统屏幕信息的程序,以弥补 FOXBASE+ 中 SAVE/ RESTORE SCREEN 命令不能处理屏幕汉字信息的不足。但使用中发现,这些程序都只能在特定的环境下运行。针对不同的显示卡和不同的汉字系统,必须对程序进行相应的修改才能适应,否则将造成屏幕闪动或根本不能运行。

我们知道,现在市面流行的汉字系统的屏幕信息显示大都是通过调用中断 10H 实现的,而 INT 10H 本身对不同显示卡具有自适应性,能自动调节内存屏幕缓存区的地址和大小。因此,可以通过调用 10H 中断来避开具体显示卡的判别,以提高程序的通用性。

下面在 BURBO PASCAL 5.0 环境下运行通过的 SAVE.PAS 和 RESTORE.PAS 程序,就是根所上面的思想编制的中西屏幕信息的保存 / 恢复程序,编译成 EXE 文件后就可以在 DOS 环境下或各种汉字系统下运行,或者在 DBASE/ FOXBASE+ 环境下通过 RUN 命令使用它们。它们的使用格式分别是:

SAVE diskfile coll row1 col2 row2

和 RESTORE diskfile

其中,diskfile 是屏幕信息存盘文件名,coll、row1、col2、row2 分别是要保存的屏幕范围:起始列、行,终止列、行的坐标值。通过调节 coll、row1、col2、row2 的值可以方便灵活地对各种中西文屏幕任意区域信息进行保存和恢复,弥补了 FOXBASE+ 中的 SAVE 和 RESTORE 命令的缺陷。

本程序在 DOS3.3、UCDOS2.1、CCDOS4.0、西出 CCDOS4.03、CCDOS2.13H 等系统下均运行通过。

program save; {屏幕信息保存程序 SAVE.PAS}

```

uses dos;
var
  reg:registers;
  fp:file of byte;
  col, row, col1, row1, col2, row2:byte;
  errcode: integer;
begin
  if paramcount,.5 then {命令行参数的合法性检查}
  begin writeln('Paramter Error!', #7); exit; end;
  val(paramstr(2), col1, errcode);{命令行中屏幕从标转换}
  val(paramstr(3), row1, errcode)
  val(paramstr(4), col2, errcode)
  val(paramstr(5), row2, errcode)
  if (col1 < 0) or (row1 < 0) or (col2 > 79) or row2 > 24) then
  {坐标的合法性检查}
  begin writeln('Screen over!', #7); exit; end;
  assign (fp, paramstr (1)); rewrite (fp);
  {$ I+}
  if IOResult < > then exit;
  write (fp, col1, row1, col2, row2);{保存屏幕坐标范围值}
  with reg do
  begin
  for row:= row1 to row2 do
  for col:= col1 to col2 do
  begin
  ah:= 2; dl:= col; dh:= row; bh:= 0
  intr($ 10,reg); {循环置光标历经要保存的点}
  ah:= 8 bh:= 0 intr($ 10,reg);

```

(下转第 42 页)

(上接第 43 页)

```
write(fp,al,ah): {保存屏幕点的 ASCII 码值和属性}
end;
ah:=2 dl:=col1;
dh:=row1; bh:=0; intr($ 10,reg) {置光标回到起始点}
close(fp)
end;
end

program restore; {屏幕信息恢复程序 RESTORE.PAS}
uses dos;
var
reg;registers;
fp:file of byte;
col, row, col1, row1, col2, row2:byte;
errcode:integer;
begin
if paramcount <> 1 then {命令行参数的合法性检查}
```

```
begin writeln('Paramte Error!', #7); exit; end;
assign(fp, paramstr(2)); reset(fp);
if IOResult <> 0 then exit;
read(fp, col1, row1, col2, row2); {读屏幕坐标范围值}
with reg do
begin
for row:=row1 to row2 do
for col:=col1 to col2 do
begin ah:=2 dl:=col; dh:=row; bh:=0;
intr($ 10, reg); {循环置光标历经要保存的点}
ah:=9; bh:=0; read(fp, al, bl); cx:=1;
intr($ 10, reg); {写各屏幕点的 ASCII 码和属性}
end;
ah:=2; dl:=col1; dh:=row1; bh:=0
intr($ 10, reg); {置光标回到起始点}
close(fp);
end;
end.
end.
```