

```

newtime = localtime(&aclock); /* 将系统时钟读入结构缓冲区 */
startsec = newtime->tm_sec;
switch(newtime->tm_wday) /* 转换星期 */
{
    case 0: strcpy(week, "星期日"); break;
    case 1: strcpy(week, "星期一"); break;
    case 2: strcpy(week, "星期二"); break;
    case 3: strcpy(week, "星期三"); break;
    case 4: strcpy(week, "星期四"); break;
    case 5: strcpy(week, "星期五"); break;
    case 6: strcpy(week, "星期六"); break;
}
while(!kbhit()) /* 循环读取和组装系统时钟字符串 */
{
    time(&aclock);
    newtime = localtime(&aclock);
    sprintf(sj, "19%d.%2d.%2d %s %d:%2d:%2d", newtime->tm_year,
        newtime->tm_mon+1, newtime->tm_mday, week, newtime->tm
hour, newtime->tm_min, newtime->tm_sec);
    currsec = newtime->tm_sec;
    if(startsec == currsec) /* 以 1 秒为周期显示系统日历时钟 */
    {
        dqx = wherex(); dqy = wherey();
        write_video(x, y, sj, 0x0a);
        goto xy(dqx, dqy);
        return;
    }
}
main()
{
    union inkey (char ch[2]; int i); c;
    for(;;){
        for(;;){
            if(kbhit() || (c.i = get_key(); break;)) /* 无按键时显示系统时钟 */
                disp_time(0, 54);
            if(c.ch[0] == 27) exit(0); /* 按 ESC 键时退出 */
        }
    }
}

```

DEBUG 功能的进一步完善

邹春贵 李文印 (长春地质学院)

DEBUG 是专门为汇编语言设计的一种调试程序。它通过单步、设置断点等方式为汇编语言程序员提供了非常有效的调试手段,是众所周知的强有力的程序调试工具。但在用它调试程序时,由于它不能把调试过程保存下来,常常让使用者感到遗憾。尽管可以利用

DOS 的改向功能把调试步骤和显示的信息输出到文件中或者打印机上。但前者只能保存信息到文件中,而不能同时在屏幕上显示;后者只能把信息打印到纸上,而不能存盘保存,均不能尽如人意。经过对 DEBUG 程序的分析,发现 DEBUG 在显示信息时,用的是往句柄 1 所对应的标准输出设备(屏幕)上写信息实现的。那么,能否在显示信息的同时,也把这些信息存盘保存呢?我们做了一下尝试,结果是可行的。具体的做法是在 DEBUG 输出信息处,把 INT 21H 改为 INT 5(其它的不用中断也可)。在 INT 5 中,第一次写信息时,首先建立文件(若文件已存在则打开,使文件定位在其末尾),以后再写信息时就不必再建立文件或打开文件了。然后显示和保存信息到文件中。如果想把键入的命令也保存下来,此部分工作也可以与上述工作同时完成。DEBUG 输入命令使用的是 INT 21H 中断的 0A 号功能实现的,可把此处的 INT.21H 也改为 INT5。

具体操作方法如下:

1. COPY DEBUG.COM CCC.DAT(Enter);

2. DEBUG CCC.DAT(Enter);

3. 用 R 命令找到文件长度及程序首址;

4. 用 S 命令寻找 INT 21H;

-s 100 3efa cd 2112AE:0756

12AE:0312 12AE:07A3

12AE:055B 12AE:082D

12AE:069D 12AE:083C

12AE:06AB 12AE:0846

12AE:06B5 12AE:0865

12AE:06C4 12AE:0879

12AE:06E7 12AE:08BA

12AE:06FD 12AE:09A1

12AE:070E 12AE:09F0

12AE:0715 12AE:09FB

5. 用 U 命令找到上面地址处 40H 送到 AH 的程序段和送 0AH 到 AH 的程序段;

-u 550

12AE:0550 IE PUHD

12AE:0551 0000 ADD[BX+1

,AL

12AE:0553 IE PUHD

12AE:0554 OE PUHC

12AE:0555 IF POPD

12AE:0556 BA2600 MOV

DX,0026

12AE:0559 B440 MOV AH,40

12AE:055BCD21 INT21

-u990

12AE:0990 0490 ADD AL,90

12AE:0992 27DAA

12AE:0993 1440 ADC AL,40

12AE:0995 27DAA

12AE:0996 247F AND AL,7F

12AE:0998 AA TOB

12AE:0999 C3 RET

12AE:099A 50 PUHAX

12AE:099B 52 PUHDX

12AE:099C B40A MOV AH,0A

12AE:099E BA5B31 MOV DX,315B

12AE:09A1 CD21 INT21

6.把相应处改为 INT 5;

-E 55C

1272:055C 21.05

-E 9A2

1272:09A2 21.05

7.把 CCC.DAT 再写回原文件;

W↓

8.把 CCC.DAT 改回原名;

RENAME CCC.DAT DEBUG.COM

9.清单一的源程序经过汇编,连接生成可执行程序;

10.在使用 DEBUG 调试程序前,先执行下面清单一的可执行程序,之后再用 DEBUG 调试程序。

清单二,是作为例子,用 DEBUG 调试清单一的执行程序时而保存在名字为 A 的文件中的信息。

清单一:

code segment

assume cs:code,ds:code

org 100h

handle dw 0

q db 0

file db 'a',0

int5: push ds

push dx

push bx

push cx

push ax

push cs

pop ds

movah,3dh

moval,2

movdx,offset file

int 21h

jnc c3

movah,3ch

xor cx,cx

movdx,offset file

int 21h

c3: movhandle,ax

movah,42h

moval,2

movbx,handle

movcx,0

movdx,0

int 21h

movax,2505h

movax,offset int05

int 21h

pop ax

pop cx

pop bx

pop dx

pop ds
int05: cmpah,0ah

jz c2

push ds

push dx

push cx

int 21h

push cs

pop ds

movbx,handle

movah,40h

pop cx

pop dx

pop ds

int 21h

iret

c2: push ds
push dx

int 21h

push cs

pop ds

movbx,handle

pop dx

pop ds

inc dx

push di

movdi,dx

movcl,[di]

movch,0

inc dx

movah,40h

int 21h

pop di

iret

begin: push cs

pop ds

movax,2505h

movdx,offset int5

int 21h

movah,31h

movdx,35h

int 21h

code ends

end begin

清单二:

-R

AX=0000 BX=0000 CX=073D DX=0000 P=0843 BP=0000

I=0000 DI=0000

D=12C5 E=12C5 S=12D5 C=12D5 IP=01B0 NV UP EI PL

NZNA PO NC

12D5:01B0 BE8401 MOV I,0184

-d cs:100 105

12D5:0100 00 00 00 61 00 1E...a...

-u 106 183

12D5:0106 52 PUH DX
 12D5:0107 53 PUH BX
 12D5:0108 51 PUH CX
 12D5:0109 50 PUH AX
 12D5:010A 0E PUH C
 12D5:010B 1F POP D
 12D5:010C B43D MOV AH,3D
 12D5:010E B002 MOV AL,02
 12D5:0110 BA0301 MOV DX,0103
 12D5:0113 CD21 INT 21
 12D5:0115 7309 JNB 0120
 12D5:0117 B43C MOV AH,3C
 12D5:0119 33C9 XOR CX,CX
 12D5:011B BA0301 MOV DX,0103
 12D5:011E CD21 INT 21
 12D5:0120 A30001 MOV [0100],AX
 12D5:0123 B442 MOV AH,42
 12D5:0125 B002 MOV AL,02
 12D5:0127 8B1E0001 MOV BX,[0100]
 12D5:012B B90000 MOV CX,0000
 12D5:012E BA0000 MOV DX,0140
 12D5:0131 CD21 INT 21
 12D5:0133 B80525 MOV AX,2505
 12D5:0136 BA4001 MOV DX,0140
 12D5:0139 CD21 INT 21
 12D5:013B 58 POP AX
 12D5:013C 59 POP CX
 12D5:013D 5B POP BX
 12D5:013E 5A POP DX
 12D5:013F 1F POP D
 12D5:0140 80FC0A CMP AH,0A
 12D5:0143 7413 JZ 0158
 12D5:0145 1E PUH D
 12D5:0146 52 PUH DX
 12D5:0147 51 PUH CX
 12D5:0148 CD21 INT 21
 12D5:014A 0E PUH C
 12D5:014B 1F POP D
 12D5:014C 8B1E0001 MOV BX,[0100]
 12D5:0150 B440 MOV AH,40
 12D5:0152 59 POP CX
 12D5:0153 5A POP DX
 12D5:0154 1F POP D
 12D5:0155 CD21 INT 21
 12D5:0157 CF IRET
 12D5:0158 1E PUH D
 12D5:0159 52 PUH DX
 12D5:015A CD21 INT 21
 12D5:015C 0E PUH C
 12D5:015D 1F POP D
 12D5:015E 8B1E0001 MOV BX,[0100]
 12D5:0162 5A POP DX
 12D5:0163 1F POP D
 12D5:0164 42 INC DX

12D5:0165 57 PUH DI
 12D5:0166 8BFA MOV DI,DX
 12D5:0168 8A0D MOV CL,[DI]
 12D5:016A B500 MOV CH,00
 12D5:016C 42 INC DX
 12D5:016D B440 MOV AH,40
 12D5:016F CD21 INT 21
 12D5:0171 5F POP DI
 12D5:0172 CF IRET
 12D5:0173 0E PUH C
 12D5:0174 1F POP D
 12D5:0175 B80525 MOV AX,2505
 12D5:0178 BA0501 MOV DX,0105
 12D5:017B CD21 INT 21
 12D5:017D B431 MOV AH,31
 12D5:017F BA3500 MOV DX,0035
 12D5:0182 CD21 INT 21

-q

PC-DOSV6.3 中 BACKUP 功能的实现

汪 林 (石家庄市胶印厂)

IBM 公司的最新 PC-DOS V6.3 操作系统中的文件备份实用工具是从 Central Point 公司移植过来的功能强大的 CPBACKUP,用它取代了沿用于数个 PC-DOS 版本中的 BACKUP 文件。但是因为广大 DOS 计算机用户多年来已习惯于使用原先的 BACKUP,感觉它使用起来简洁、方便、快速。另外,若用 CPBACKUP.EXE 备份的文件在低版本中不能使用 RESTORE 程序来恢复,存在兼容性问题。所以,许多用户希望在 PC-DOSV6.3 中还能继续使用原先的 BACKUP 功能。然而,若在 PC-DOSV6.3 下直接运行 DOS 低版本,如 DOSV5.0 中的 BACKUP. EXE 程序,屏幕将出现下面错误提示:Incorrect DOS version。

上面提示说明 DOS 版本不匹配。那么怎样才能解决这个问题,使得在 6.3 环境下正常使用 BACKUP 功能,广大读者自然会想到使用 DOS 的外部命令 SETVER,这种方法使用十分简单,只要在 DOS 系统启动配置文件 CONFIG.SYS 中加入一条语句:

```
DEVICE=C:/DOS/SETVER.EXE
```

然后在 DOS 提示符下键入命令:

```
SETVER C:/DOS/BACKUP.EXE 5.0
```