如何在C语言中直接使用XMS

祝建中 (杭州师范学院)

摘要:本文提出了一种在高级语言编写的应用程序中,如何直接利用扩充内存块 *XMB* 来存储数据,以改善程序的运行环境,提高可用内存容量和程序运行速度的方法。并以 *C* 语言为例,给出了该方法的实现技术。

一、基本概念

在XMS中,扩充内存由上位存储块UMB、高位存储区HMA和扩充内存块XMB三部分组成。PC机的640K边界至1M边界间的384K内存空间通常是预留给DOS系统、视频缓冲、BIOS的,但常会有部分空闲区,把扩充内存映射至这些区,就可以利用这些区的地址访问扩充内存。这些区域就称为UMB。而HMA是指扩充内存的第一个64KB区域(实际为65520字节)。它能在实模式且A20线被激活的状态下被CPU访问。扩充内存中除去HMA及被映射成UMB以外的部分,就称为XMB。

在高版本DOS中使用HIMEM.SYS就能无冲突地利用扩充内存。通常DOS的大部分被装入HMA,利用EMM386可以管理UMB,从而可将一些设备驱动程序及驻留程序装入UMB,以得到更多的基本内存。但占扩充内存大部分的XMB,往往只是被简单地用作虚拟盘、磁盘CACHE等。若能在应程序,特别是普遍使用的C语言编写的应用程序中,直接利用这些数倍于基本内存容量的XMB来存储数据,就可以打破实模式下的"640KB壁垒",成倍地增加可用内存容量,改善程序的运行环境,提高运行速度。这对无法利用EMM386将扩充内存映射成UMB来增加可用基本内存量的286,所得的效益更大。本文以C语言为例,讨论如何应用这种技术的方法。

二、XMS功能详解

XMS驱动程序HIMEM.SYS提供了驱动程序基本信息、HMA管理、A20线管理、XMB管理和UMB管理五组功能。了解及正确地使用这些功能,是直接利用扩充内存编程的基础。表1列出了功能号与功能的对应关系。因为HMA已让DOS占用(至少45KB),A20线主要用于HMA读写,UMB可由EMM386管理用来装入驱动程序和驻留程序,故HMA和UMB中可利用的存储空间已不多,所以我们主要讨论有关使用SMB的功能。

表1 XMS功能表

功能号	功 能
000H	取XMS版本号,HMA状态
001H	申请高位内存区域HMA
002H	释放高位内存区域HMA
003H	全程启用地址线A20
004H	全程停用地址线A20
005H	局部启用地址线A20
006H	局部停用地址线A20
007H	查询地址线A20状态
008H	查询可用的扩充内存容量
009H	分配XMB
00AH	释放XMB
00BH	移动内存块
00CH	锁住XMB
00DH	解锁XMB
00EH	查询XMB的句柄信息
00FH	重新分配XMB
010H	申请上位内存块UMB
011H	释放上位内存块UMB

1.000H功能

入口参数:AH=00H

出口参数:AX XMS的版本号,BCD码表示

BX XMS驱动程序的版本号,用BCD码表示

DX HMA存在标态。0000H,HMA不存在;0001H,HMA存在。

BL 错误代码(见表2,下同)

2.008H功能

入口参数:AH=08H

出口参数:AX 以KB计的量大未用XMB

DX 以KB计的未用XMB之总和

BL 错误代码

3.009H功能

入口参数:AH=09H

DX 以KB计的要分配的XMB大小

出口参数:AX0000H失败,0001H成功

DX 被分配块的句柄

BL 错误代码

© 中国科学院软件研究所 http://www.c-s-a.org.cn

4.00AH 功能

入口参数：AH＝0AH
　　DX 欲释放块的句柄
出口参数：AX0000H 成功
　　BL 错误代码

表 2　BL 错误代码

代码号	含　义
080H	功能未实现
081H	检测到 VDISK 设备
082H	A20 出现错误
090H	不存在 HMA
091H	HMA 正被占用
092H	申请的字节长度小于 HMAMIN
093H	HMA 未分配
0A0H	所有的 XMB 部已分配
0A1H	所有的 XMB 句柄都已被占用
0A2H	句柄无效
0A3H	无效的源句柄
0A4H	无效的源偏移值
0A5H	无效的目的句柄
0A6H	无效的目的偏移值
0A7H	长度无效
0A8H	移动有重叠
0A9H	奇偶校验错
0AAH	块未加锁
0ABH	块被加锁
0ACH	块加锁计数溢出
0ADH	加锁失败

5.00BH 功能

入口参数：AH＝0BH
　　DS：SI 指向内存块移动结构(见表 3)的指针
出口参数：AX0000H 失败，0001H 成功
　　BL 错误代码

本功能不仅能将数据块在基本内存和扩充内存之间传送，而且可将块在基本内存或扩充内存中传送。表 3 中，要传送的长度必须为偶数，基本内存的句柄号为 0，扩充内存块的句柄号应为由功能 09H 所获者，各偏移值中的高 16 位是段地址，低 16 位是偏移量。

6.00CH 功能

入口参数：AH＝0CH
　　DX 要锁往的块句柄
出口参数：AX0000H 失败，0001H 成功
　　DX：BX 块首的 32 位线性地址
　　BL 错误代码

表 3　内存块移动结构

偏 移 量	储存的信息含义
00H～03H	需传送的安节数
04H～05H	源块句柄
06H～09H	32 位源偏移值
0AH～0BH	目的块句柄
OCH～0FH	32 位目的偏移值

7.00DH 功能

入口参数：AH＝0DH
　　DX 要解锁的块句柄
出口参数：AX0000H 失败，0001H 成功
　　BL 错误代码

8.00EH 功能

入口参数：AH＝0EH
　　DX 块句柄
出口参数：AX0000H 失败，0001H 成功
　　BH 块的加锁计数值
　　BL 如果成功，为系统中还未用的句柄数；
　　如果失败，为错误代码。

9.00F 功能

入口参数：AH＝0FH
　　BX 以 KB 计的块的新长度
　　DX 要改变大小的块的句柄
出口参数：AX0000H 失败，0001H 成功
　　BL 错误代码

三、在 C 中的实现

要检查 XMS 是否存在及获取 XMS 驱动程序的控制地址，需要调用 2FH 号多路服务中断，具体参数如下：

1.04300H 子功能

判断 XMS 是否存在。

入口参数：AH＝43H
　　AL＝00H
出口参数：AL＝08H，则 XMS 存在，否则不存在。

2.04310H 子功能

获取 XMS 驱动程序入口指针地址。

入口参数：AH＝43H
　　AL＝10H
出口参数：ES 入口指针段地址 4
　　BX 入口指针偏移量

© 中国科学院软件研究所 http://www.c-s-a.org.cn

程序给出了一个示例，它将整个屏幕的信息存入XMB，然后再从XMB读取并用其恢复屏幕。它不但节约了基本内存，而且恢复速度与直接写屏相同。程序在AST386及多种286、386兼容机上用Borland C++3.1或TturboC2.0编译运行通过。

因为BorlandC++3.1的BC.EXE是完全的DPMI-(DOS Protection Mode Interface)宿主软件，集成开发环境将占有所有的XMS，所以上述程序必须生成EXE文件后退出BC在DOS环境下运行。如果是286，可用Turbo C2.0。因为程序中含有汇编码，所以要用带选项-B的命令行编译器TCC编译，并且要调用Turbo Assembler。当然，CONFIG.SYS中必须包含XMS驱动程序HIMEM.SYS。

```
/*        程序XMSDEMO.C在C中直接使用XMS示例*/
#include "XMSdevice.h"
int WriteToXMS(char far *add, long int count, unsigned handle)
{
XMBstruct xmb;
unsigned long seg, off;
seg=FP_SEG(add0;
off=FP_OFF(add);
xmb.count=count;
xmb.SourceHandle=0;
xmb.SourceOfs=(seg<<16)+off;
xmb.DestinHandle=handle;
xmb.Destin0fs=0;
return CXMS_move(*xmb));
}
int ReadFromXMS(char for* add, long int count, unsigned handle)
{
XMBstruct xmb;
unsigned long seg, off;
seg=FP_SEG(add);
off=FP_OFF(add)
xmb.count=count;
xmb.SourceHandle=handle;
xmb.SourceOfs=0;
xmb.DestinHandle=0;
xmb.DestinOfs=(seg<<16)+off;
return(XMS_move(&xmb));
}
int main()
{
char far *sb;
unsigned handle1, handle2; /*两个XMB的句柄/
unsigned long add1, add2; /*两个XMB的32位线性地址*/
int i;
sb=(char far *)MK_FP(0xb800,0x0000); /*VGA模式3的视频缓冲区首址*/
clrscr();
if (XMS_exist()==FALSE)(/*测试XMS是否存在*/
  printf("Extended memroy is NOT present/n");
  exit (0);
}
else
 {
 window(1,1,80,25);
 gotoxy(25,11);
 cprintf("EDM0 for useing XMS,1 screen/n");
 gotoxy(25,14);
 cprintf("在C中直接使用XMS示例,第1屏/n");
 XMS_alloc(8,&handle1);/*申请第一个大小为8KB的XMB,以储存第一屏*/
 writeToXMS(sb,8000,handle1)/*将第一屏信息写入第一个XHB*/
 XMS_lock(handle1,add1);/*锁住第一块XMB*/
 getch();
 textcolor(WHITE);
 gotoxy(25,11);
 cprintf("DEM0 for useing XMS,2 screen/n");
 gotoxy(25,11);
 gotoxy(25,14);
 cprintf("在C中直接使用XMS示例,第2屏/n");
 XMS_alloc(8,handle2);申请第二个大小为8KB的XMB,以储存第二屏*/
 writeToXMS(sb,8000,handle2);/*将第二屏信息写入第二个XMB*/
 XMS_lock(handle2,add2);/*锁住第二块XMB*/
 getch();
 for (i=;i<3;i++){
  /*依次从XMS中取出第一、第二屏信息来恢复屏幕,重复3次*/
  ReadFROMxms((char far *)sb,8000,handle1);
  getch();
  ReadFromXMS((char fer *)sb,8000,handle2);
  getch();
  }
 XMS_unlock(handle1);     /*第一个XMB解锁*/
 XMS_free(handle1);       /*释放第一个XMB*/
 XMS_unlock(handle2);     /*第二个XMB解锁*/
 XMS_free(handle2);       /*释放第二个XMB*/
 textcolor(LIGHTGRAY);
 textbackground(BLACK);
 clrscr();
 }
return0;
}
```

参考文献

[1]程渝荣，DOS 5.0存储管理特性及工具，中国计算机用户，1992年第12期

[2]夏旭东，MS DOS 5.0下的系统优化，计算机世界，1993

© 中国科学院软件研究所 http://www.c-s-a.org.cn