无操作系统环境下硬盘参数获取

罗冰
南京理工大学计算机系，南京（210094）
E-mail: luobing4365@163.com

摘　要：硬盘是最常用的存储介质之一。在各种安全应用的场合，常常需要获取硬盘的详细参数，但是大部分文献只研究了在操作系统下获取硬盘信息的理论和方法。本文研究了在无操作系统的情况下获取硬盘参数的方法，并对 IDE 硬盘和 SATA 硬盘的不同进行了阐述，在此基础上给出了具体实现方案和程序流程。

关键词：硬盘；int 13h；CHS；LBA；ECh
中图分类号： T P319

1 引言

硬盘是一个半导体与机械的集合体。自从 1956 年的 IBM 350 问世以来，盘片技术、电机技术都有了质的飞跃。一直以来，我们都是采用 ATA/IDE 并行接口的方式进行硬盘数据的传输。Intel 在 2000 年的 IDF 会上提出了 SATA 接口，到现在，SATA 接口的硬盘几乎完全替代了 IDE 接口的硬盘了。

在很多的 IT 项目中，都需要获取硬盘的信息，比如软件加密等。如果是在操作系统层面，使用操作系统提供的 API 能够满足项目要求。当前很多获取硬盘信息的实现方法都是在操作系统层面进行的，理论和实现方法都已经比较完善。在无操作系统支持的环境中，只能使用 BIOS 中断和 I/O 指令访问硬件，此时获取硬盘信息是不小的挑战。另外的一个难题是现在流行的 SATA 硬盘与传统的 IDE 硬盘不同，其 I/O 访问端口不是固定不变的，必须通过 BIOS 提供的信息来确定其端口值。

本文阐述了如何在无操作系统的环境下访问硬盘，并给出了使用 I/O 指令获取硬盘信息的方法和代码。

2 底层访问硬盘的方法

在无操作系统的情况下，有两种方法可以访问硬盘，其一是使用 BIOS 提供的中断。另外一种是直接使用 I/O 指令访问。以下分别说明。

2.1 硬盘的逻辑结构

BIOS 中断提供了 int 13h 来访问硬盘，常使用的有 CHS 模式和 LBA 模式两种逻辑结构。在了解如何通过硬盘中断访问硬盘前，必须了解 CHS 模式和 LBA 模式的区别，这两中模式分别对于于标准硬盘中断和扩展硬盘中断。

2.1.1 CHS 模式

硬盘的 CHS 模式是指 Cylinder/Head/Sector 模式，很久以前, 硬盘的容量还非常小的时候, 硬盘采用的结构与软盘类似，也就是硬盘盘片的每一条磁道都具有相同的扇区数。由此产生了所谓的 3D 参数 (D isk G eometry)，即磁头数(Heads), 柱面数(Cylinders), 扇区数(Sectors per track)，以及相应的寻址方式。

其中：磁头数(Heads) 表示硬盘总共有几个磁头,也就是有几面盘片，最大为 256 (用 8 个二进制位存储)；柱面数(Cylinders) 表示硬盘每一面盘片上有几条磁道，最大为 1024(用 10 个二进制位存储)；扇区数(Sectors per track) 表示每一条磁道上有几个扇区，最大为 63 (用

6 个二进制位存储)。每个扇区一般是 51 2 个字节，理论上讲这不是必须的，但是当前基本上所有硬盘都是采用这样的方式运行[1]。

使用 CHS 方式访问硬盘是有限制的，即最大只能访问到 8G，这就是常说的标准 BIOS 限。为了突破访问限制，在其后的发展中提出了 LBA 模式[1]。

2.1.2 LBA 模式

LBA，全称为 Logic Block Address（即扇区的逻辑块地址）。在 LB A 模式下，设置的柱面、磁头、扇区等参数并不是实际硬盘的物理参数。在访问硬盘时，由控制器把由柱面、磁头、扇区等参数确定的逻辑地址转换为实际硬盘的物理地址。

其实对编程人员来说，相较于 CHS 模式，可以认为柱面数可以超过 1024 即可，这样在使用中断的时候理解起来比较方便。LBA 与 CHS 的转换可以由以下公式给出：

LBA=(C-Cs) * Ph *Ps+(H-Hs) *Ps+(S-Ss) [2]

其中，C 表示当前柱面号，H 表示当前磁头号，S 表示当前扇区号，Cs 表示起始柱面号，Hs 表示起始磁头号，Ss 表示起始扇区号，Ps 表示每磁道的扇区数，Ph 表示每柱面的磁道数。

2.2 硬盘 BIOS 中断

BIOS Int 13H 调用是 BI OS 提供的磁盘基本输入输出中断调用，它可以完成磁盘(包括硬盘和软盘)的复位、读写、校验、定位、诊断、格式化等功能。在其规范发展的过程中，分别提供了标准的 int 13h 和扩展的 int 13h。如前所叙，标准的 int 13h 使用 CHS 寻址模式，扩展的 int 13h 使用 LBA 寻址模式。

2.2.1 标准的 int 13h

标准的 int13h 提供的服务一般从功能 0 到功能 25。通过中断 13h 访问硬盘服务，所有的功能都要求在 DL 中用一个号来指定要使用的驱动器。将 DL 中的第 7 位置高表示正在访问硬盘，否则就是在访问软盘。将 DL 设为 80h,可访问硬盘驱动器 0；同样地，当 DL 设为 81h 时可访问驱动器 1。因本文主要讨论获取硬盘参数，下面给出读取硬盘驱动器参数地 API[1]：

```
//读磁盘驱动器参数
//入口： AH =8；DL=驱动器号(80h+从零开始地硬盘驱动器号)
//返回：CF=0：
        AH=0；AL=未定义
        CH=最大磁柱号（10 位磁柱号的低 8 位）
        CL：第 6、7 位=磁柱号的第 8、9 位；第 0~5 位=最大扇区号(1~63)
        DH=最大磁头号(0~255)
        DL=驱动器数目
      CF=1：
        如果至少激活了一个驱动器：
          AH=7；AL=CX=DX=0
        如果没有硬盘存在：
          AH=1；AL=BX=CX=DX=ES=0
```

可以看出标准的 int13h 只能简单的获取硬盘的 CHS 参数而已，无法满足更多的要求。

2.2.2 扩展的 int 13h

设计扩展 I nt13H 接口的目的是为了扩展 BI OS 的功能，使其支持多于 1024 柱面的硬盘，以及可移动介质的琐定、解锁及弹出等功能。扩展 int 1 3h 的功能号从 41h 开始，功能 48h 提供的功能是获取扩展驱动器参数[1]。

首先给出扩展驱动器参数的数据结构：

```
typedef struct DriveParametersPackettag
{
    unsigned int InfoSize;          // 数据包尺寸 (26  字节)
    unsigned int Flags;             // 信息标志
    long   int Cylinders;           // 磁盘柱面数
    long   int Heads;               // 磁盘磁头数
    long   int SectorsPerTrack;     // 每磁道扇区数
    long   int Sectors1;            // 磁盘总扇区数
  lo   ng  in  t Sectors2;
  u   nsigned int SectorSize;       // 扇区尺寸 (以字节为单位)
    unsigned int DPTE_ofs;          // 指向 DPTE 的指针
  unsi   gned int DPTE_seg;
}DriveParametersPacket;
```

请注意，unsigned int 是 16 位长，long int 为 32 位长。以上给出的是在使用扩展 int 13h 的功能 48h 时必须用到的数据结构。API 的说明如下：

```
//获取硬盘扩展驱动器参数
//入口:
    AH = 48h
    DL = 驱动器号
    DS:DI = 返回数据缓冲区地址(即指向上述的扩展驱动器参数数据结构)
//返回:
    CF = 0, AH = 0 成功 DS:DI 驱动器参数数据包地址
    CF = 1, AH = 错误码
```

注意在扩展驱动器参数中的 DPTE 指针，只有在 InfoSize 设置为 30 或者更大，并且硬盘支持增强磁盘驱动器（现在一般的硬盘都支持）时，才能够获得，否则此值会设为 FFFF:FFFFh 指示无效。以下为 DPTE 指向的参数表[3]：

Offset	Type	Description
0-1	Word	I/O port base address
2-3	Word	Control port address
4	Byte	Head register upper nibble
5	Byte	BIOS Vendor Specific.
6	Byte	IRQ information bits 0-3 IRQ for this drive bits 4-7 0
7	Byte	Block count for ATA READ/WRITE MULTIPLE commands
8	Byte	DMA information bits 0-3 DMA channel bits 4-7 DMA type
9	Byte	PIO information bits 0-3 PIO type bits 4-7 0
10-11	Word	BIOS selected hardware specific option flags
12-13h	Word	Reserved, shall be 0
14	Byte	11h, revision level of this table.
15	Byte	Checksum, 2's complement of the 8 bit unsigned sum of bytes 0-14

图 1 DPTE(Device parameter table extension)

图中给出的Byte是8位，Word为16位。我们需要注意的是偏移0和2给出的I/O Base address(I/O基址)和Control port address(控制端口地址)。在实际应用中，IDE主硬盘的I/O基址一般是1F0h，控制端口地址为3f6h；从盘为170h和376h。但是SATA硬盘一般不是这么确定的，我们就可以通过DPTE给出的参数表获取当前访问硬盘的I/O基址和控制端口地址。

从扩展int 13h给出的资料可以看出，功能48h提供的硬盘参数还是比较有限，比如硬盘序列号就无法获取。如果需要获取硬盘更详细的信息，就必须使用I/O指令来获取了。

2.3 通过 I/O 指令访问硬盘

2.3.1 端口简介

为便于下面的说明，我们简要介绍硬盘控制器的端口（IO_BASE_ADDR表示硬盘控制器的I/O基址）[3]：

表 1 硬盘控制器 I/O 端口说明

I/O端口：IO_BASE_ADDR+偏移			
偏移	对应		描述
0	读/写	数据端口寄存器	主机和设备的数据传输通过这个寄存器进行
1	读	错误码寄存器	最后一次命令执行的状态信息
	写	写补偿	驱动忽略此寄存器
2	读/写	扇区计数	定义数据传输的扇区数量
3	读/写	首扇区编号	包含传输的第一个扇区的ID
4	读/写	柱面低8位	如果访问操作跨越柱面，其值自动增1，以反映当前柱面
5	读/写	柱面高8位	与柱面低8位一起反映当前柱面
6	读/写	驱动器/磁头选择	选定驱动器

7	读	状态寄存器	命令执行完成之后驱动器更新这个寄存器反映当前设备状态
	写	命令寄存器	一旦命令写入寄存器，设备就开始执行这个命令

2.3.2 获取硬盘参数的 I/O 指令

查阅ATA的标准协议，可知获取硬盘信息的命令是ECh。往命令寄存器发送ECh后，可以通过数据端口寄存器获取相应的驱动器信息。驱动器信息由512个字节给出，其主要内容有（以字节偏移计算）：

偏移0：介质信息

偏移2：总逻辑磁柱数

偏移20：硬盘序列号码，共20个ASCII字符

偏移46：固件版本

因篇幅原因，就不一一列出了，可以参照ATA协议来获取所需要的信息。但是从上面的讨论中我们可以明显看出，通过I/O指令获取的磁盘信息比BIOS中断获取的信息要更为详细。下面我们给出获取硬盘信息的实现代码。

3 代码实现

以下给出的代码均在Borland C++ 3.1上编译通过。

3.1 获取硬盘控制器 I/O 基址

现在流行的SATA端口与传统的IDE硬盘不同，其端口是浮动的。幸好BIOS中断提供了通过DPTE获取硬盘I/O基址的方法，我们的代码中采用的就是这种方法。代码如下[3]：

```
//无操作系统下可以操作指针的宏
#define MAKE_FP(seg,ofs)  ((v  oid FAR *)(((unsigned long)(seg) << 16) | (unsigned int)(ofs)))
#define FP_SEGM(fp)  ((u  nsigned int)((unsigned long)((void FAR *)(fp)) >> 16))
#define FP_OFFS(fp)          ((unsigne    d int)(fp))
//获取硬盘控制器I/O基址
//入口参数：0~3  硬盘序号，一般第一个硬盘为0
//出口参数：I/O基址 [4]
unsigned int getPort(unsigned char num)
{
  unsigned int far *dpte;
  DriveParametersPacket drivepage;
  unsigned int addr;

  addr=(unsigned int)(&drivepage);
  drivepage->InfoSize=70;
  asm{
    mov ah,48h
    mov dl,80h
```

```
    add dl,num
    mov si,addr
    int 13h
    jc _err_
  }
  dpte=MAKE_FP(drivepage->DPTE_seg,drivepage->DPTE_ofs);
  ruturn dpte[0];
_err_: return 0;
}
```

3.2 获取硬盘详细信息

按照上面的讨论，给出使用I/O命令ECh获取硬盘信息的代码：

```
//获取硬盘详细信息
//入口参数: I/O基址
//出口参数: 0:函数工作成功，信息存放在全局变量中  1:函数工作失败
unsigned char infoHD[512];        //存放硬盘信息的缓冲区
unsigned char getHDinfo(unsigned int IO_BASE)
{
  unsigned int buf_seg,buf_ofs,count;
  buf_seg=FP_SEGM(infoHD);
  buf_ofs=FP_OFFS(infoHD);
  asm{
    pusha
    push  ES

    mov dx,IO_BASE
    add dx,6
    mov al,0e0h
    out dx,al                        //首先发送e0,设置硬盘为LBA模式
    out 0ebh,al
    out 0ebh,al
    inc dx                           //命令寄存器
    mov al,0ech
    out dx,al                        //给命令寄存器发送EC命令
  }
  count=10000;
  wait_for_hd_nbusy:
  count--;
  if(count==0)
    return 1;
  asm{
```

```
  out 0ebh,al
  out 0ebh,al
  in al,dx
  test al,80h
  jnz wait_for_hd_nbusy          //等待硬盘驱动器不忙，BUSY=0?
}
count=10000;                     //计数限
wait_for_hd_driver:
count--;
if(count==0)
  return 1;
asm{
  out 0ebh,al
  out 0ebh,al
  in al,dx
  test al,40h
  jz wait_for_hd_driver          //等待驱动器已经准备好，DRDY=1?
}
count=10000;
wait_for_data_ready:
count--;
if(count==0)
  return 1;
asm{
  out 0ebh,al
  out 0ebh,al
  in al,dx
  and al,08h
  jz wait_for_data_ready
  cli
  cld
  mov cx,256
  mov dx,IO_BASE
  mov di,buf_ofs
  mov ax,buf_seg
  push  ax
  pop ES
  REP   INSW                     //从数据端口取得512个字节
  sti
  pop ES
  popa
```

```
    }
    return 0;
}
```

4 结束语

本文给出了在无操作系统支持下获取访问硬盘的方法，并分析了SATA硬盘与传统IDE硬盘的访问方法的异同。其主要不同在于硬盘控制器的I/O基址浮动，可以通过BIOS提供的信息来确定硬盘控制器的I/O基址，然后通过ECh命令可以获取完整的硬盘信息。最后给出了实现代码，对底层应用和安全的研究方向有一定的参考作用。

参考文献

[1] Frank.van.Gilluwe. 《PC 技术内幕》[M].精英科技.北京：中国电力出版社，2001.
[2] 戴士剑，涂彦辉. 《数据恢复技术》[M]，北京：电子工业出版社，2005.3。
[3] Peter.T.Mclean.《Information Technology-AT Attachment with Packet Interface-6》[EB/OL].
[4] http://www.t13.org, 2001.3。
[5] 沈美明，温冬婵. 《80x86 汇编语言程序设计》[M] .北京：清华大学出版社，2001.

Get the parameters of HD without operation system support

Luo Bing
Department of Computer Science, Nanjing University of Science and Technology, Nanjing (210094)

Abstract

HD is a common data storage medium.We often need to obtain detailed parameters of HD in variety of computer security applications. However, the majority of the literature studied only in the operating system access to the HD of the theories and methods of information .The paper describes the method to get parameters of HD whitout operation system support, and gives the different of programming in IDE and SATA. Finally, given the realization of specific programs and code.

Keywords: HD; int 13h; CHS; LBA; ECh