哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

汇编语言程序设计

第11&12讲：BIOS及DOS功能调用

裴文杰

第11&12讲：BIOS及DOS功能调用

BIOS与DOS简介

键盘I/O

显示器I/O

串行通信口I/O

文件存取I/O

wenjiecoder@outlook.com

BIOS与DOS简介

键盘I/O

显示器I/O

串行通信口I/O

文件存取I/O

wenjiecoder@outlook.com

BIOS与DOS简介

基本概念

(1) BIOS

IBM PC系列机在只读存储器ROM中固化有一组外部设备驱动与管理软件，组成PC机基本输入输出系统(Basic Input/Output System，BIOS),它处于系统软件的最低层，又称ROM BIOS。

BIOS主要包括以下一些功能：

①系统自检及初始化

例如，系统加电启动时对硬件进行检测、对外部设备进行初始化、设置中断向量、引导操作系统等。

②系统服务

为操作系统和应用程序提供系统服务，这些服务主要与I/O设备有关，如读取键盘输入、显示等。为了完成这些操作，BIOS必须直接与I/O设备打交道，它通过端口与I/O设备之间传送数据，使应用程序脱离具体的硬件操作。

③硬件中断处理　　提供硬件中断服务程序

计算机系统软件就是利用这些基本的设备驱动程序与管理软件，完成各种功能操作。

wenjiecoder@outlook.com

BIOS与DOS简介

(2) DOS

磁盘操作系统（Disk Operating System），他是早期PC机的重要操作系统之一，主要完成对文件、设备、内存的管理。主要包括三个模块：IBMBIO.com，IBMDOS.com，COMMAND.com，其功能分别为：

(a)IBMBIO.COM：DOS在ROM BIOS的基础上开发了一组输入输出设备处理程序，是DOS与ROM BIOS的接口，这组程序称为IBMBIO.COM。

(b)IBMDOS.COM：在IBMBIO.COM的基础上，DOS还开发有文件管理程序和一些处理程序，称为IBMDOS.COM。

(c)COMMAND.COM：DOS的命令处理程序。

COMMAND.COM与IBMBIO.COM、IBMDOS.COM一起构成了基本DOS系统。

wenjiecoder@outlook.com

BIOS与DOS简介

DOS 21H 5 BIOS 17H 0

- BIOS BIOS 17H 2 DOS
- DOS BIOS DOS

wenjiecoder@outlook.com

BIOS和基本DOS的作用

- 用户可通过使用BIOS和DOS系统提供的这些功能模块子程序（中断子程序调用），来编制直接管理和控制计算机硬件设备的底层软件（主要是完成I/O操作）。
- 用户不必深入了解有关设备的电路和接口，只须遵照DOS规定的调用原则即可使用。

用户编程原则

- 尽可能使用DOS的系统功能调用。
- 在DOS功能不能实现情况下，考虑用BIOS功能调用。
- 在DOS和BIOS的中断子程序不能解决问题时，使用IN/OUT指令直接控制硬件。

wenjiecoder@outlook.com

BIOS与DOS简介

调用BIOS/DOS功能子程序的基本方法：

BIOS/DOS的每个功能都对应着一个中断服务程序

采用int中断的BIOS/DOS中断属于软件中断

中断调用指令格式

```
INT    n                ; n称为中断类型号
```

其中：

① DOS中断：n= 20H~3FH

② BIOS中断： n= 5~1FH

③ 自由中断：n=40H~FFH（用户可自定义）

wenjiecoder@outlook.com

DOS中断和BIOS中断使用方法：

[illegible] AH [illegible] AL [illegible]，

除此之外，还需要在CPU [illegible]。

一般来说，调用DOS或BIOS功能时，有以下几个步骤：

1 [illegible]

2 [illegible] AH

3 [illegible] AL

4 [illegible] DOS [illegible] BIOS

5 [illegible]

例：可调用DOS功能调用来设置中断向量，把由AL指定的中断类型的中断向量DS：DX放入中断向量表。

调用参数：DS：DX=中断向量、AL=中断类型

功能号：AH=25H

执行：INT 21H

DOS功能调用

类型号	中断功能	类型号	中断功能
20H	程序结束	21H	请求DOS功能调用
22H	结束地址	23H	中止(Ctrl-Break)处理
24H	关键性错误处理	25H	磁盘顺序读
26H	磁盘顺序写	27H	程序结束且驻留内存
28H	DOS内部使用	29~2EH	DOS内部保留
2FH	DOS内部使用	30~3FH	DOS内部保留

wenjiecoder@outlook.com

BIOS与DOS简介

与DOS相比，BIOS是在更低的层次上为用户提供系统服务

CPU中断、8259A中断、BIOS中断

类型号	中断功能	类型号	中断功能
00 H	被零除	11 H	设备检测
01 H	单步	12 H	存储容量
02 H	不可屏蔽	13 H	磁盘I/O
03 H	断点	14 H	通信I/O
04 H	溢出	15 H	盒式磁带I/O
05 H	打印屏幕	16 H	键盘I/O
06 H	保留	17 H	打印机I/O
07 H	保留	18 H	ROM BASIC
08 H	日时钟	19 H	引导
09 H	键盘	1A H	日时钟
0A H	保留	1B H	Ctrl-Break
0B H	串口2	1C H	定时器报时
0C H	串口1	1D H	显示器参数
0D H	硬盘	1E H	软盘参数
0E H	软盘	1F H	图形字符扩展
0F H	打印机	40 H	保留给软盘
10 H	显示器	41 H	硬盘参数

wenjiecoder@outlook.com

BIOS与DOS简介

DOS调用与BIOS调用两者的异同：

DOS功能调用在更高层次上提供了与BIOS类同的功能。

DOS调用与BIOS调用两者的区别

- 调用BIOS中断程序比调用DOS的复杂一些，但运行速度快，功能更强；
- DOS功能调用只适用于DOS环境，而
  BIOS功能调用不受任何操作系统的约束；
- 某些功能只有BIOS具有。

wenjiecoder@outlook.com

BIOS与DOS简介

例：将一个ASCII字符显示于屏幕的当前光标所在位置。

使用DOS的21号中断的2号功能：

```
        MOV   DL, '?';要显示的字符送入DL
MOV       AH, 2
        INT              21H              ;调用21H软中断
```

使用BIOS的10H中断的0EH号功能：

```
        MOV   AL, '?';要显示的字符送入AL
        MOV              AH,0EH           ;功能号送入AH
        INT              10H              ;调用10H软中断
```


wenjiecoder@outlook.com

BIOS与DOS简介

键盘I/O

字符码与扫描码

BIOS键盘中断

DOS中断调用

显示器I/O

串行通信口I/O

文件存取I/O

wenjiecoder@outlook.com

键盘I/O

字符码与扫描码：

键盘是计算机最基本的一种输入设备，用来输入信息，以达到人机对话的目的。键盘主要由3种基本类型的键组成：

字符数字键：A~Z, 0~9, 以及常用符号%, $, #等。

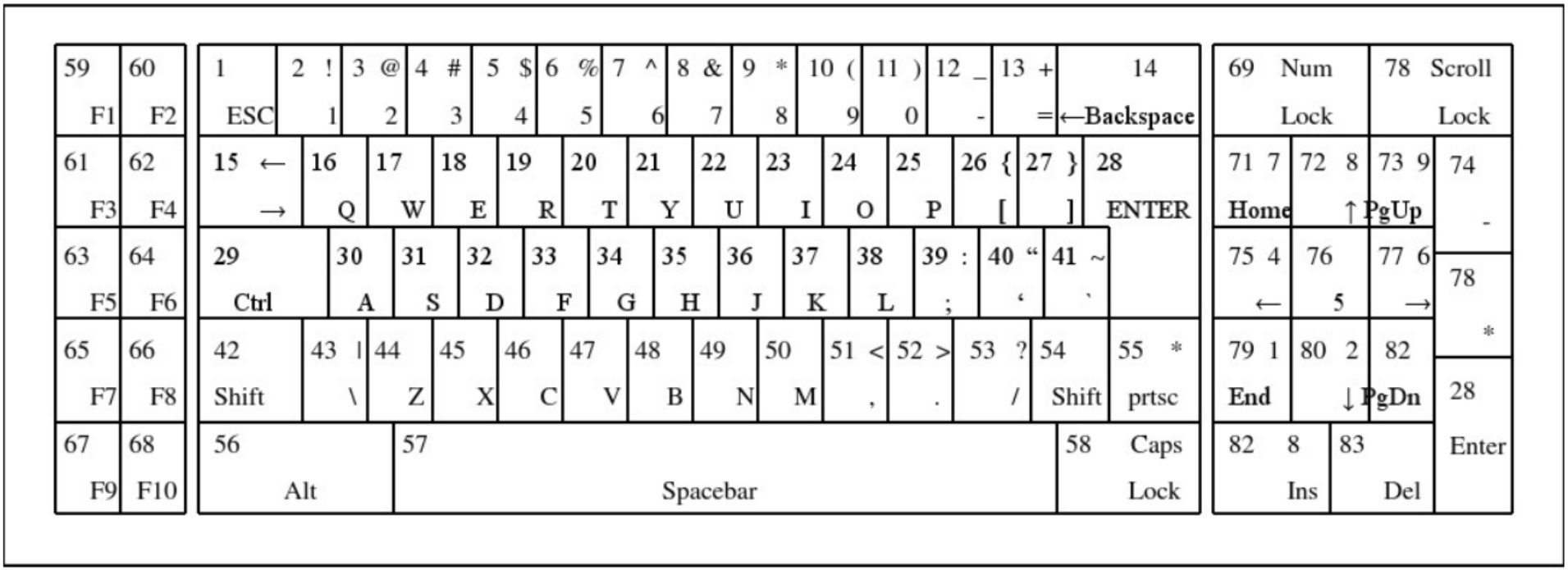
扩展功能键：如Home、End、Backspace、Delete、Insert、PgUp、PgDown以及功能键F1~F10等。

组合使用的控制键：如Alt, Ctrl, Shift等。

字符数字键给计算机传送一个ASCII码字符，而扩展功能键产生一个动作，使用组合控制键能改变其他键所产生的字符码。

wenjiecoder@outlook.com

键盘I/O

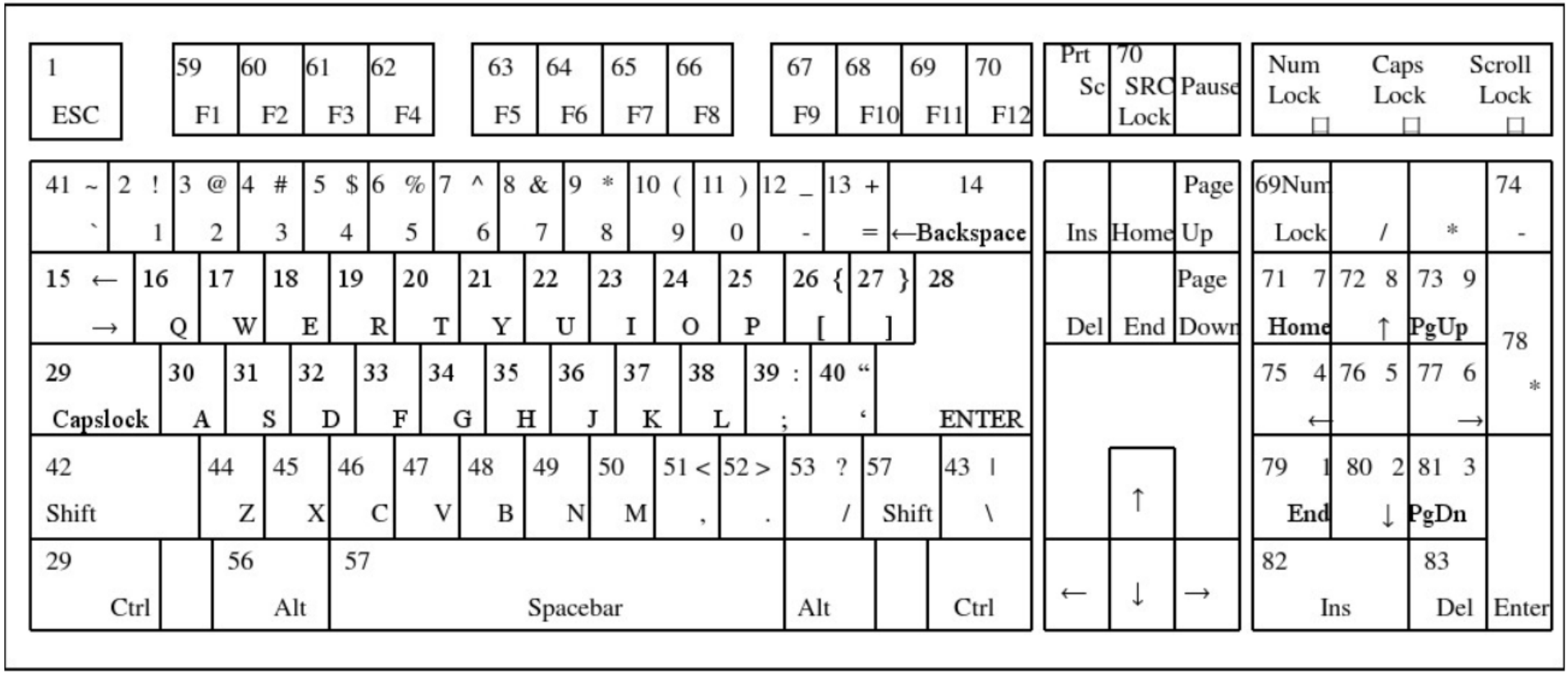


83键键盘的键位布局和扫描码

83键是笔记本或平板电脑键盘的mini键盘的规格。

wenjiecoder@outlook.com

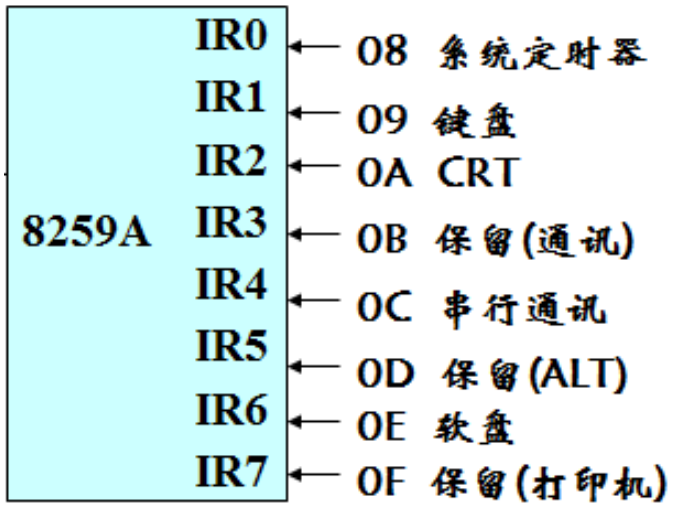
键盘I/O

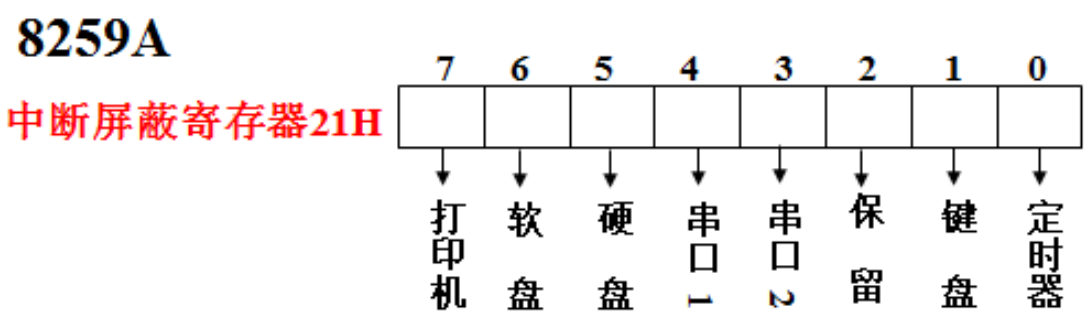


美式101键键盘的键位布局和扫描码

键盘I/O

- PC机系列的键盘触点电路按16行×8列的矩阵来排列，用单片机Intel8048来控制对键盘的扫描。
- 按键的识别采用行列扫描法，即根据对行线和列线的扫描结果来确定闭合键的位置，这个位置值称为按键的扫描码，通过数据线将8位扫描码送往主机。
- 当在键盘上“按下”或“放开”一个键时，如果键盘中断是允许的（8259A 21H端口的第一位等于0），就会产生一个类型9 的中断，并转入到BIOS的键盘中断处理程序。





wenjiecoder@outlook.com

键盘I/O

- 该处理程序从8255可编程外围接口芯片的输入端口读取一个字节，这个字节的低7位是按键的扫描码。最高位为0或者为1，分别表示键是“按下”状态还是“放开”状态。按下时，取得的字节称为通码，放开时取得的字节称为断码。如ESC键按下取得的通码为01H（00000001B），放开ESC键时会产生一个断码81H（10000001B）。

- BIOS键盘处理程序将取得的扫描码转换成相应的字符码，大部分的字符码是一个标准的ASCII码；没有相应ASCII码的键，如Alt和功能键（F1~F10），字符码为0；还有一些非ASCII码键产生一个指定的操作，如PRTSC。

wenjiecoder@outlook.com

键盘I/O

转换成的字符码以及扫描码存储在BIOS数据区的键盘缓冲区KB_BUFFER中：

```
Buffer_Head   DW   ?              ;键盘缓冲区头指针
Buffer_ Tail    DW   ?              ;键盘缓冲区尾指针
KB_Buffer      DW   16 DUP(?)    ;键盘缓冲区的缺省长度为16个字
KB_Buffer_End   Label   Word
```

键盘缓冲区是一个先进先出的循环队列。虽然缓冲区的本身长度为16个字，但出于判断“队列满”的考虑，它最多只能保存15个键盘信息。当缓冲区满时，系统将不再接受按键信息，而会发出“嘟”的声音，以示要暂缓按键。当Buffer_Head=Buffer_Tail时，说明缓冲区为空，表示无键盘输入。

可以用BIOS中断，也可以用DOS中断和键盘通信。

wenjiecoder@outlook.com

键盘I/O

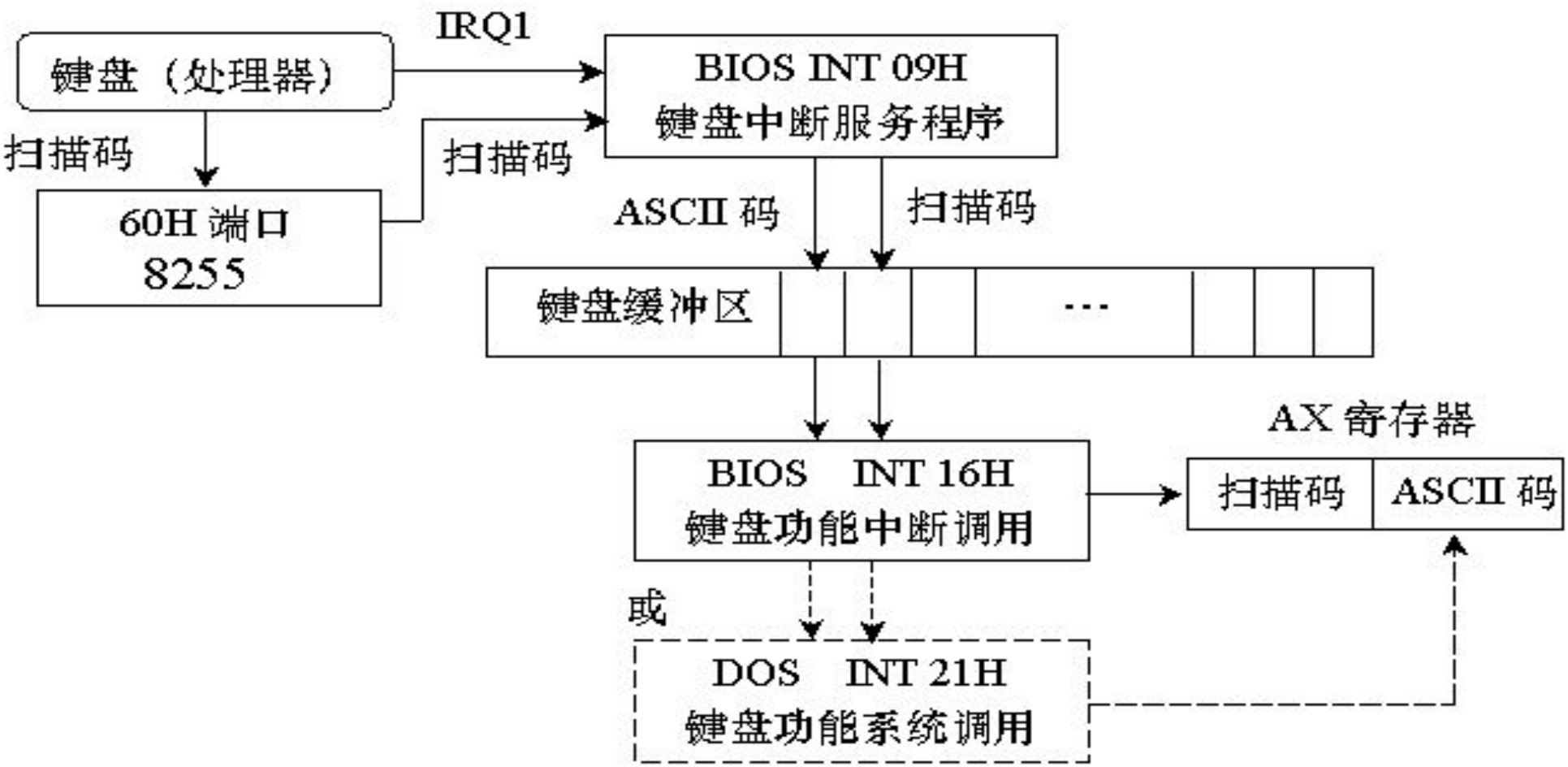


图 5.13 键盘中断处理流程

键盘中断处理流程

wenjiecoder@outlook.com

键盘I/O

键盘中断：09H vs. 16H?

- 09H：硬件中断；16H：软件中断。
- BIOS的INT 09H和INT 16H中断处理程序是一对相互配合的程序，其中09H向键盘缓冲区写入，16H从键盘缓冲区读出。
- 处理时机不同：09H是有键按下的时候产生硬件中断并触发；16H是应用程序调用的时候起作用。

wenjiecoder@outlook.com

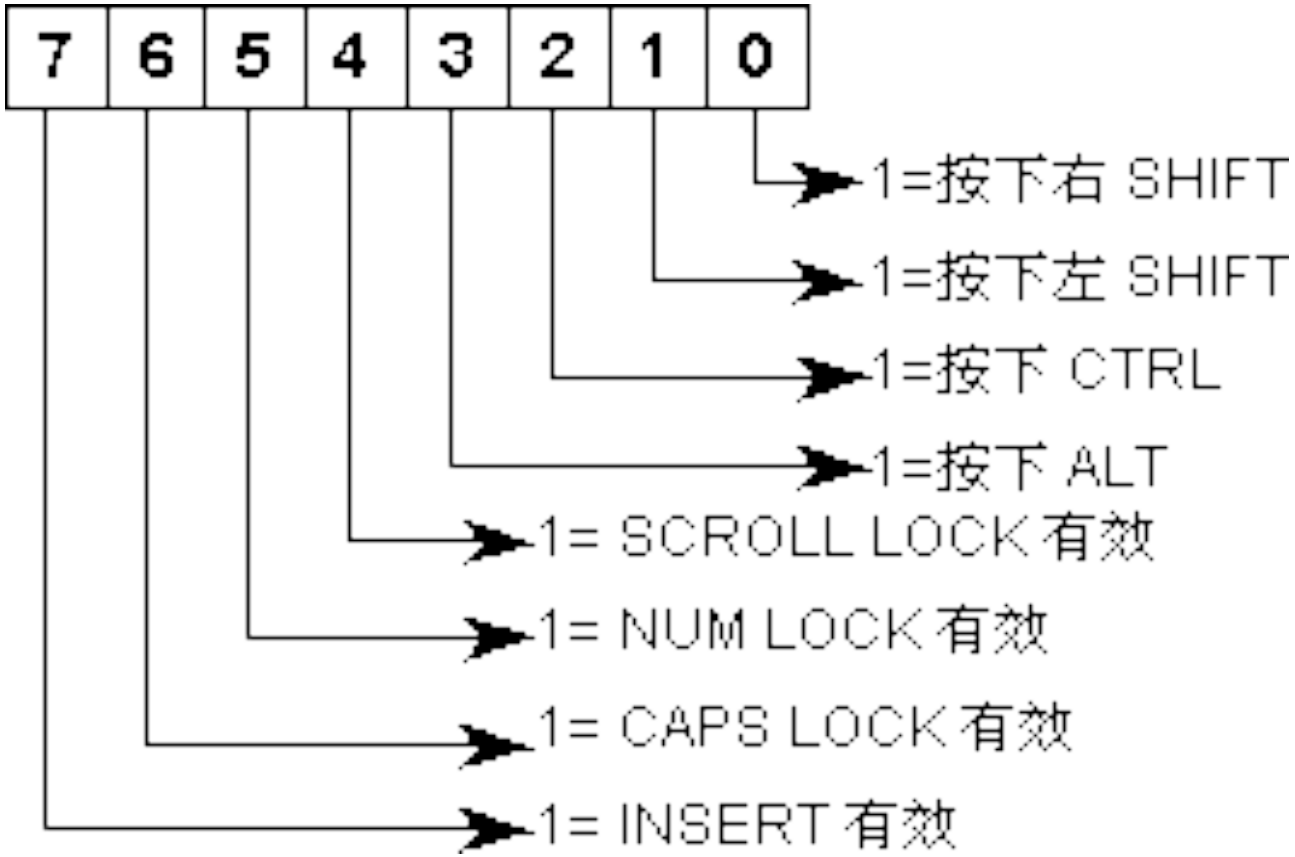
键盘状态字

在计算机键盘上除了可输入各种字符(字母、数字和符号等)的按键之外，还有一些控制键(如：Ctrl、Alt、Shift等)、双态键(如：Num Lock、Caps Lock等)和特殊请求键(如：Print Screen、Scroll Lock等)。

键盘中的控制键和双态键是非打印按键，它们是起控制或转换作用的。当使用者按下控制键或双态键时，系统要记住其所按下的按键。为此，在计算机系统中，特意安排的一个字节来标志这些按键的状态，我们称该字为键盘状态字节（KB_Flag）。

wenjiecoder@outlook.com

键盘状态字节的各位含义如下图所示。



INT

16H的2号功能，可以把键盘状态字节（KB_Flag）回送到AL，其中1表示按下

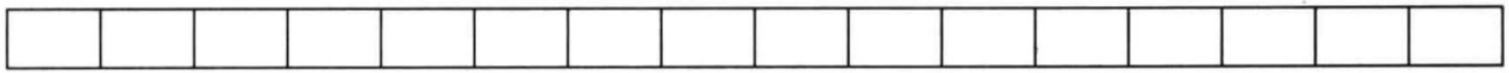
int 9 键盘中断

下面，我们通过下面几个键：

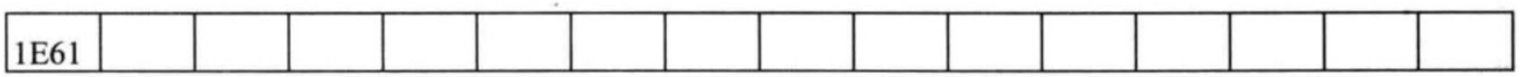
A、B、C、D、E、Shift_A、A

的输入过程，简要地看一下 int 9 中断例程对键盘输入的处理方法。

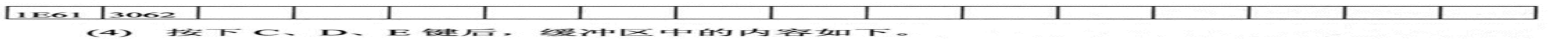
(1) 初始状态下，没有键盘输入，键盘缓冲区空，此时没有任何元素。



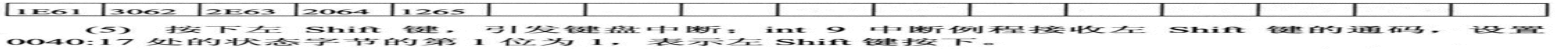
(2) 按下 A 键，引发键盘中断；CPU 执行 int 9 中断例程，从 60h 端口读出 A 键的通码；然后检测状态字节，看看是否有 Shift、Ctrl 等切换键按下；发现没有切换键按下，则将 A 键的扫描码 1eh 和对应的 ASCII 码，即字母“a”的 ASCII 码 61h，写入键盘缓冲区。缓冲区的字单元中，高位字节存储扫描码，低位字节存储 ASCII 码。此时缓冲区中的内容如下。



(3) 按下 B 键，引发键盘中断；CPU 执行 int 9 中断例程，从 60h 端口读出 B 键的通码；然后检测状态字节，看看是否有切换键按下；发现没有切换键按下，将 B 键的扫描码 30h 和对应的 ASCII 码，即字母“b”的 ASCII 码 62h，写入键盘缓冲区。此时缓冲区中的内容如下。

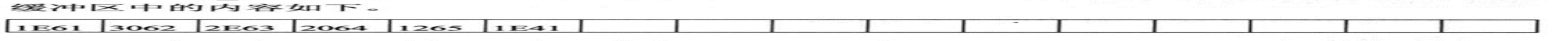


(4) 按下 C、D、E 键后，缓冲区中的内容如下。



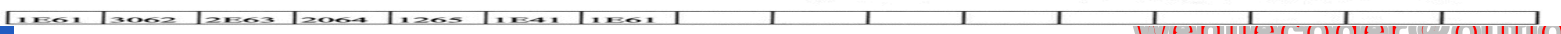
(5) 按下左 Shift 键，引发键盘中断；int 9 中断例程接收左 Shift 键的通码，设置 0040:17 处的状态字节的第 1 位为 1，表示左 Shift 键按下。

(6) 按下 A 键，引发键盘中断；CPU 执行 int 9 中断例程，从 60h 端口读出 A 键的通码；检测状态字节，看看是否有切换键按下；发现左 Shift 键被按下，则将 A 键的扫描码 1Eh 和 Shift_A 对应的 ASCII 码，即字母“A”的 ASCII 码 41h，写入键盘缓冲区。此时缓冲区中的内容如下。



(7) 松开左 Shift 键，引发键盘中断；int 9 中断例程接收左 Shift 键的断码，设置 0040:17 处的状态字节的第 1 位为 0，表示左 Shift 键松开。

(8) 按下 A 键，引发键盘中断；CPU 执行 int 9 中断例程，从 60h 端口读出 A 键的通码；然后检测状态字节，看看是否有切换键按下；发现没有切换键按下，则将 A 键的扫描码 1Eh 和 A 对应的 ASCII 码，即字母“a”的 ASCII 码 61h，写入键盘缓冲区。此时缓冲区中的内容如下。



(4) 按下 C、D、E 键后，缓冲区中的内容如下。

1E61	3062	2E63	2064	1265											

(5) 按下左 Shift 键，引发键盘中断；int 9 中断例程接收左 Shift 键的通码，设置 0040:17 处的状态字节的第 1 位为 1，表示左 Shift 键按下。

(6) 按下 A 键，引发键盘中断；CPU 执行 int 9 中断例程，从 60h 端口读出 A 键的通码；检测状态字节，看看是否有切换键按下；发现左 Shift 键被按下，则将 A 键的扫描码 1Eh 和 Shift_A 对应的 ASCII 码，即字母“A”的 ASCII 码 41h，写入键盘缓冲区。此时缓冲区中的内容如下。

1E61	3062	2E63	2064	1265	1E41										

(7) 松开左 Shift 键，引发键盘中断；int 9 中断例程接收左 Shift 键的断码，设置 0040:17 处的状态字节的第 1 位为 0，表示左 Shift 键松开。

(8) 按下 A 键，引发键盘中断；CPU 执行 int 9 中断例程，从 60h 端口读出 A 键的通码；然后检测状态字节，看看是否有切换键按下；发现没有切换键按下，则将 A 键的扫描码 1Eh 和 A 对应的 ASCII 码，即字母“a”的 ASCII 码 61h，写入键盘缓冲区。此时缓冲区中的内容如下。

1E61	3062	2E63	2064	1265	1E41	1E61									

wenjiecoder@outlook.com

BIOS键盘中断

类型 16H

的中断提供了基本的键盘操作，它的中断处理程序包括3个不同的功能，分别根据AH寄存器中的子功能号来确定。

(1) AH=0：从键盘读字符到AX寄存器中。其中AL=字符码，AH=扫描码

如果键盘缓冲区为空，等待键盘输入

(2) AH=1：读键盘缓冲区字符到AX寄存器中，并置ZF标志位。

若ZF=0，则AL=字符码，AH=扫描码

如果键盘缓冲区为空，ZF=1，不等待键盘输入

若ZF=1，则表示缓冲区空

(3) AH=2：读取键盘状态字节。(AL=键盘状态字节）。

wenjiecoder@outlook.com

键盘I/O

例：从键盘读1个字符，并将其对应的字符码和扫描码打印出来。

```
…
mov   ah, 0
int    16h
mov   bx, ax
BHTOAL  ;字符码
mov   cx, ax
mov   al, bh
BHTOAL  ;扫描码
…
```

BHTOAL宏：
将AL中的值转换成ASCII码。

```
BHTOAL   MACRO
         MOV     AH, AL
AHHN     MACRO
         LOCAL   AHHN1
         MOV     CL, 4
         SHR     AH, CL
         CMP     AH, 10
         JC      AHHN1
         ADD     AH, 7
AHHN1:   ADD     AH, 30H
         ENDM
ALLN     MACRO
         LOCAL   ALLN1
         AND     AL, 0FH
         CMP     AL, 10
         JC      ALLN1
         ADD     AL, 7
ALLN1:   ADD     AL, 30H
         ENDM
         AHHN
         ALLN
         ENDM
```

读取特殊功能键的状态

```
code segment
        assume cs:code
        mov  ah, 2
        int     16h
        mov   ah, 4ch
        int     21h
   code  ends
end
```


wenjiecoder@outlook.com

DOS键盘功能调用（INT 21H）

无

DX

wenjiecoder@outlook.com

键盘I/O

1）单字符输入：

INT 21H的1号功能，AL=输入字符

例9-2
接收键盘输入并对其进行测试。

```
get_key:mov   ah, 1
        int    21h
        cmp  al , 'y'
       je     yes
        cmp  al, 'n'
       je      no
       jne   get_key
```

例9-3
检测键盘输入的字符是否是回车键。

```
wait_here: mov  ah, 7
           int    21h
           cmp   al, 0dh
           jne   wait_here
```

键盘I/O

如果程序要求能接收功能键或数字组合键必须进行两次DOS功能调用:

第一次回送00

第二次回送扫描码

例9-4
程序显示一个菜单，要求用户通过输入F1、F2、F3来选择，其他按键则产生错误信息。

```
            …
            mov  ah,7
            int  21h
            cmp  al,0
            je   get_char
            jmp  error
get_char:
            mov  ah,7
            int  21h
            cmp  al, 3bh        ;F1?
            je   option1
            cmp  al, 3ch        ;F2?
            je   option2
            cmp  al, 3dh        ;F3?
            je   option3
            jmp  error
            …
error:
            …
```

2）输入字符串

INT 21H的10号功能，DS：DX=缓冲区首地址。

- 缓冲区的第一个字节保存最大字符数（逻辑上限是255），由用户设定，若输入的字符数大于此数，PC机会发出“嘟嘟”声，光标不再移动。
- 第二个字节是实际输入字符的个数。这个数据由功能10自动填入。
- 两个字节之后就是用户输入的字符串，以回车键结束（也会占用一个字节）。
- 因此缓冲区的大小应为：最大字符数（包括回车）+2

```
MaxLen  DB  32
ActLen   DB  ?
String    DB  32  DUP (? )
```

```
LEA    DX,  MaxLen
MOV    AH,  0AH
INT    21H
```


wenjiecoder@outlook.com

```
data segment
    smax  db 21
    sact    db ?
    stri     db 21 dup(?)
data ends
code segment
    assume cs:code,ds:data
Start:mov ax,data
     mov ds,ax
     lea    dx, smax
     mov ah, 0ah
     int    21h
     mov ah,4ch
     int    21h
 code ends
End start
```

例9-5　输入字符串程序

缓冲区的定义方式:

(1) smax db 21
    sact db ?
    stri db 21 dup(?)

(2) smax db 21
        db ?
        db 21 dup(?)

(3)smax db 21, ?, 21 dup(?)

(4)smax db 21, 22 dup(?)

wenjiecoder@outlook.com

3）清除键盘缓冲区

Int

21的功能0ch能清除键盘缓冲区，然后执行在AL中指定的功能。AL中指定的功能可以是1，6，7，8或0AH。

```
...
mov   ah, 0ch
mov   al, 08h
int    21h
```

使用0CH的好处是可以避免因为偶然超前输入的字符而出现错误。

4）检验键盘状态

INT 21H的0B号功能可以检验一个键是否被按动，如果按下一个键，则AL=0FFH，否则AL=0。无论哪种情况都将继续执行下一条语句。

例 编写按任意键结束程序的程序段

```
Wait:  …
       mov   ah, 0bh
       int    21h
       inc    al
       jnz    Wait
       ret
```

BIOS与DOS简介

键盘I/O

显示器I/O

字符属性

BIOS显示中断

DOS显示功能调用

串行通信口I/O

文件存取I/O

wenjiecoder@outlook.com

显示器I/O

显示器通过显示适配器与PC机相连。显示适配器也称为显卡，是计算机和显示器的接口。

显示器两种显示方式：

- 文本方式

  将屏幕分成若干行和列，在每个网格位置上显示字符。

- 图形方式

  将屏幕分成m*n的点阵，在每个点的位置显示像素。

wenjiecoder@outlook.com

显示器I/O

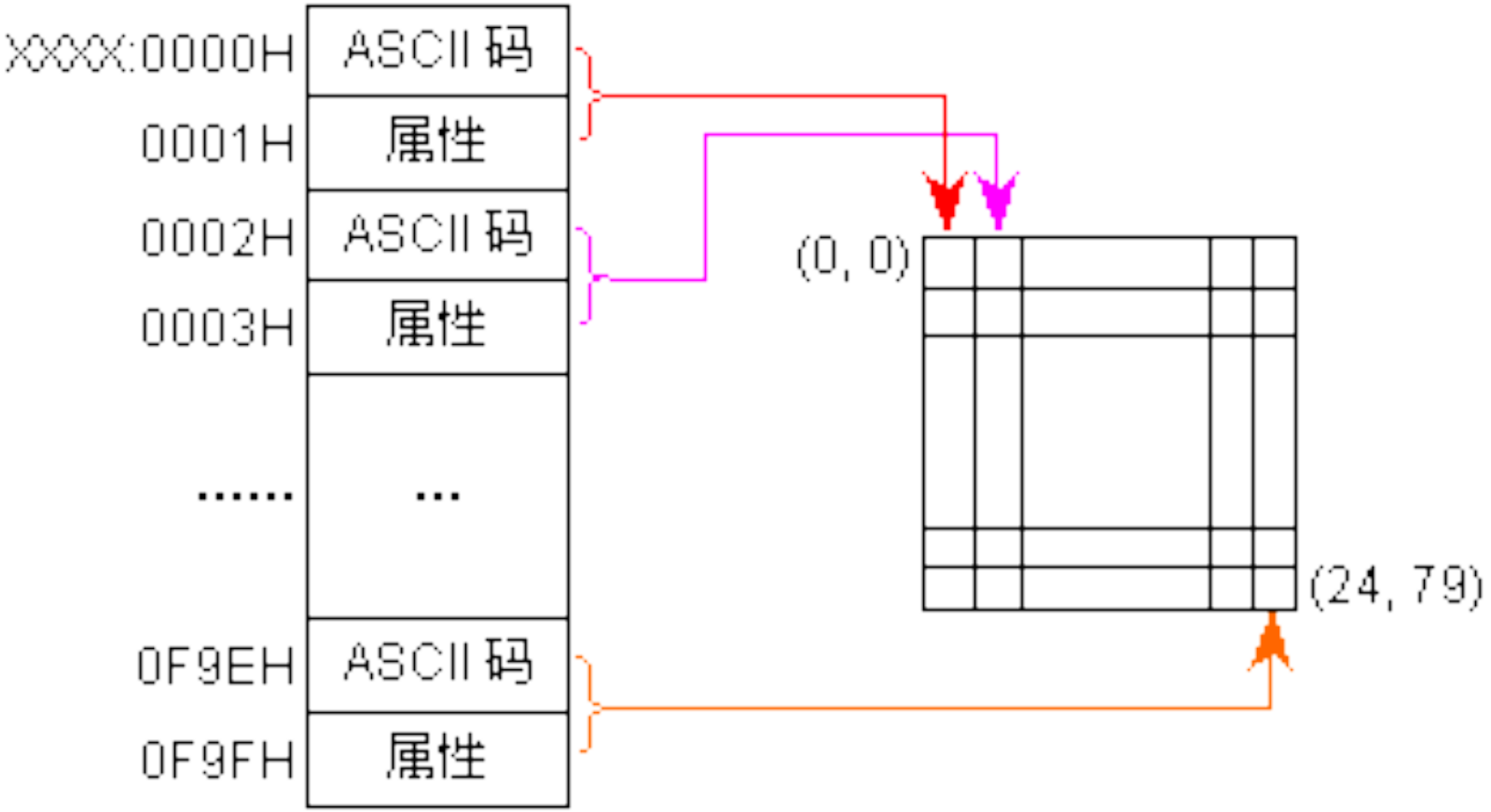
文本显示方式 字符属性

- 显示器的屏幕通常划分为行和列的一个二维系统，显示适配器就在行和列组成的网格位置上显示字符。
- 每个字符都是以矩形块形式显示的。在BIOS ROM中存有多种不同大小的字符集，主要的显示字符集大小为：8×8(标准)、8×14和8×16等。
- 在常用的文本显示模式下，屏幕被划分成25行，每行可显示80个字符，所以，每屏最多可显示2000(80×25)个字符。
- 为了便于标识屏幕上的每个显示位置，我们就用其所在行和列来表示之，并规定：屏幕的左上角坐标为(0, 0)，右下角坐标为(24, 79)。

wenjiecoder@outlook.com

显示器I/O

- 对应屏幕上的每个字符位置，主存空间都有相应的存储单元与之对应，因此可说是显示屏幕是“存储器的映像”。
- 对应显示屏幕上的每个字符，在存储器中由连续的两个字节表示，
  一个字节表示ASCII码，另一个字节保存字符的属性。



显示器I/O

1、单色字符显示：对单色显示，字符的属性确定了该字符的显示方式，如字符是否闪烁、是否高亮度、是否反向显示等。

7	6 5 4	3	2 1 0
闪烁位	背景色	亮度色	前景色
0=正常显示 1=闪烁显示	000=黑 111=白	0=正常亮度 1=加强亮度	000=黑 111=白

属性值（二进制）	属性值（十六进制）	显示效果
00000000	00	无显示
00000001	01	黑底白字，下划线
00000111	07	黑底白字，正常显示
00001111	0F	黑底白字，高亮度
01110000	70	白底黑字，反相显示
10000111	87	黑底白字，闪烁
11110000	F0	白底黑字，反相闪烁

如：黑底白字闪烁显示，可设置属性为87H（10000111）

不显示字符：00H

显示器I/O

2、彩色字符显示：

显示彩色字符时，属性字节可以选择显示字符的前景颜色和背景颜色。

- 前景颜色有16种可以选择，背景颜色有8种可以选择。
- 闪烁和亮度只应用于前景。(BL为闪烁位，I为亮度位)

彩色字符显示属性字节

位号	7	6 5 4	3 2 1 0
属性字节	BL	R G B	I R G B
	闪烁选择	背景颜色	前景颜色

显示器I/O

背景颜色组合

RGB	颜色
000	黑
001	蓝
010	绿
011	青
100	红
101	品红
110	棕
111	白

前景颜色组合

IRGB	颜色	IRGB	颜色
0000	黑	1000	灰
0001	蓝	1001	浅蓝
0010	绿	1010	浅绿
0011	青	1011	浅青
0100	红	1100	浅红
0101	品红	1101	浅品红
0110	棕	1110	黄
0111	白	1111	强度白

3. 显示存储器

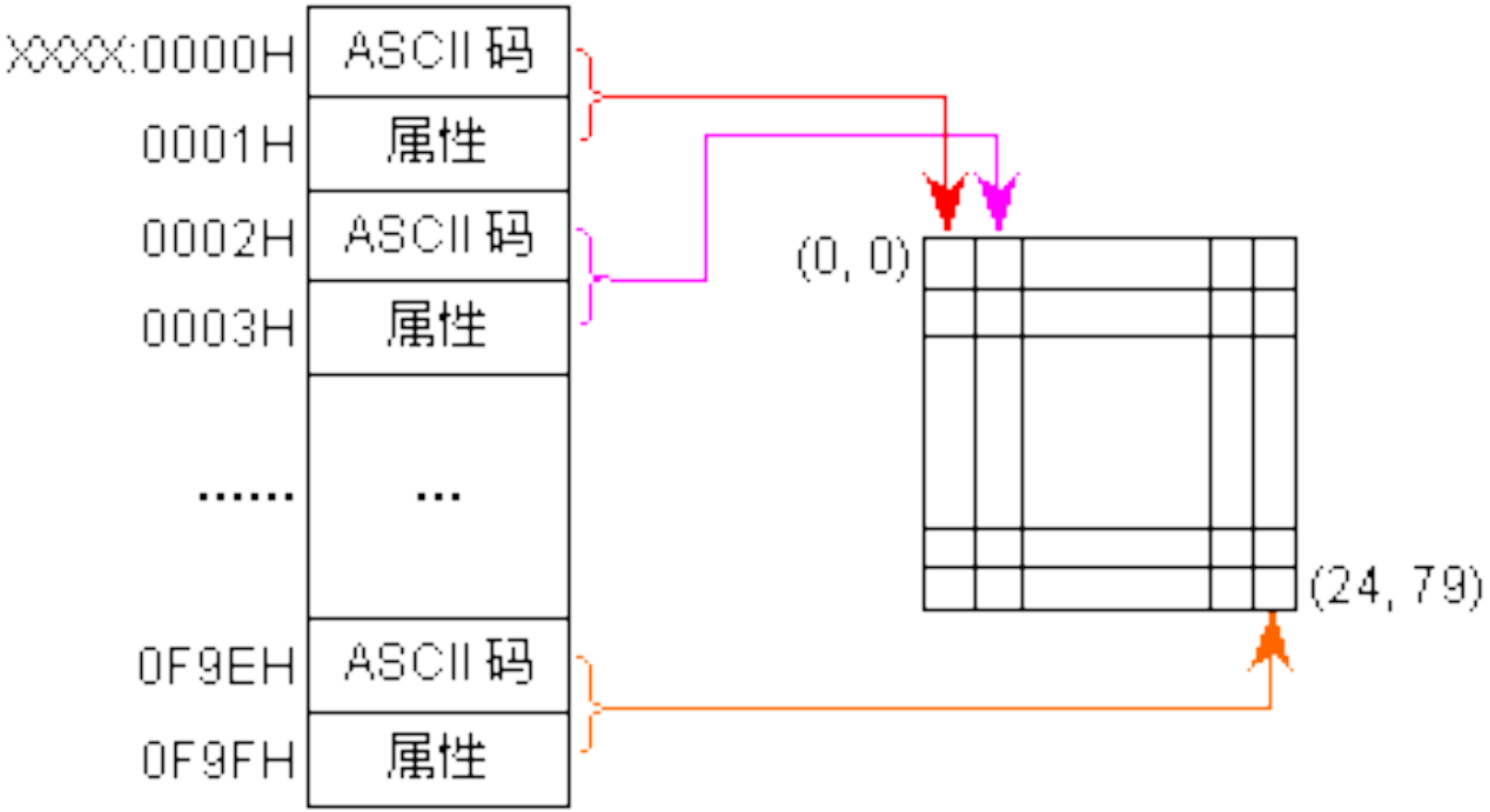
对于所有的显示适配器，文本方式下显示字符的原理是一样的，所不同的是显存的起始地址不同，对于单色显示适配器MDA，显存的起始地址为B000:0000；对于CGA、EGA、VGA是B800:0000。

在25 X 80的文本显示方式下，屏幕可有2000个字符位置，因每个字符需要用两个字节，所以每屏显存容量需要4K。若显存有16K，则可以保存4屏的显示字符数据。

对VGA的80列显示方式，0页的起始地址是B800:0000，1页的起始地址是B800:1000，2页的起始地址为B800:2000，3页的起始地址为B800:3000。

内存中0xB8000(MDA是B0000)到0xBFFFF
这段物理地址被映射到显存，以便于加速cpu和显卡之间的数据传输，也就是说，cpu写这些物理地址，就如同写到了显存一样，可以控制显示内容。具体内存映射机制这里不讲
。

显示器I/O



屏幕上某一字符在显存中的偏移地址可由下列公式算出：

Char_Offset=Page_Offset+(row*width+column)*byte

byte是表示一个字符所用的字节数，这里：byte=2。

wenjiecoder@outlook.com

显示器I/O

BIOS显示中断调用

10H中断调用为显示器中断，共有17种功 能。下面列出几种主要功能的使用情况。

(1) 设置光标类型（1号功能）

入口参数：AH=1（功能号），CH=光标开始行，CL=光标结束行。

出口参数：无。根据CX给出光标的大小。

(2) 设置光标位置（2号功能）

入口参数：AH=2（功能号），BH=页号，DH=行号，DL=列号。

出口参数：无。根据DX确定了光标位置。

(3) 读当前光标位置（3号功能）

入口参数；AH=3（功能号），BH=页号。

出口参数：DH=行号，DL=列号，CX=光标大小。

wenjiecoder@outlook.com

显示器I/O

例：置光标的类型：开始行 为1，结束行 为7，并把它设置到第五行、第六列。

```
CODES SEGMENT
   ASSUME CS:CODES
START:
   mov ch,1
   mov cl,7
   mov ah,1
   int  10h
   mov dh,4
   mov dl,5
   mov bh,0
   mov ah,2
   int  10h
   mov ah,1     ;输入字符并显示
   int  21h
   MOV AH,4CH
   INT 21H
CODES ENDS
   END START
```

设置光标类型（1号功能）

入口参数：AH=1（功能号），CH=光标开始行，CL=光标结束行。

只用CH、CL的低4位，

若CH的第4位为1，光标不显示。

出口参数：无。

根据CX给出光标的大小。

设置光标位置（2号功能）

入口参数：AH=2（功能号），BH=页号，DH=行号，DL=列号。

出口参数：无。

根据DX确定了光标位置。

wenjiecoder@outlook.com

（4）初始窗口或向上滚动（6号功能）

入口参数：AH=6，AL=上滚行数，CX=上滚窗口左上角的行、列号。DX=上滚窗口右下角的行、列号。BH=空白行的属性。

出口参数：无。当滚动后，底部为空白输入行。 如果AL=0，清除屏幕。

（5）初始窗口或向下滚动（7号功能）

入口参数：AH=7，AL=下滚行数，CX=下滚窗口左上角的行、列号。DX=下滚窗口右下角的行、列号。BH=空白行的属性。

出口参数：无。当滚动后，顶部为空白输入行。如果AL=0，清除屏幕。

（6）读当前光标位置的字符与属性（8号功能）

入口参数：AH=08H，BH=页号。

出口参数：AL为读出的字符，AH为字符属性。

（7）在当前光标位置写字符和属性（9号功能）

入口参数：AH=9，BH=页号，AL=字符的ASCII码，BL=字符属性，CX=写入字符重复次数。

出口参数：无。

例：在屏幕中心建立一个20列宽和9行高的小窗口。在小窗口的最下一行输入字符，满一行就向上卷动。


```
data segment
    esc_key    equ  1bh  ；ESC的ASCII码
    win_ulc    equ  30   ；左上角列号
    win_ulr    equ  8    ；左上角行号
    win_lrc    equ  50   ；右下角列号
    win_lrr    equ  16   ；右下角行号
    win_width equ  20
data ends
code segment
     assume cs:code,ds:data
start: mov ax, data
     mov, ds, ax
again: mov ah,2
     mov   dh,win_lrr
     mov   dl,win_ulc
     mov   bh,0
     int     10h
     mov    cx,win_width
get_char:
     mov    ah,1
     int     21h
     cmp al, esc_key
     jz exit
     loop   get_char
     mov    ah,6
     mov    al,1
     mov   ch,win_ulr
     mov   cl,win_ulc
     mov   dh,win_lrr
     mov   dl,win_lrc
     mov   bh,7
     int     10h
     jmp    again
exit:  mov ah,4ch
     int       21h
  code     ends
end
```

设置光标位置（2号功能）

入口参数：**AH=2**（功能号），**BH=页号，DH=行号，DL=列号。**

出口参数：无。根据DX确定了光标位置。

初始窗口或向上滚动（6号功能）

入口参数：**AH=6，AL=上滚行数，CX=上滚窗口左上角的行、列号。DX=上滚窗口右下角的行、列号。BH=空白行的属性。**

出口参数：无。当滚动后，底部为空白输入行。

例9.13.asm

例：在品红背景下，显示5个浅绿色闪烁的星号

```
code segment
      assume cs:code
start:
      mov         ah,9
      mov         al,'*'
      mov         bh,0
      mov         bl,0dah
      mov         cx,5
      int         10h
      mov         ah,4ch
      int         21h
  code   ends
end start
```

例9.16.asm

在当前光标位置写字符和属性（9号功能）

入口参数：AH=9，BH=页号，AL=字符的ASCII码，BL=字符属性，CX=写入字符次数。

出口参数：无。

背景颜色组合

RGB	颜色
000	黑
001	蓝
010	绿
011	青
100	红
101	品红
110	棕
111	白

前景颜色组合

IRGB	颜色	IRGB	颜色
0000	黑	1000	灰
0001	蓝	1001	浅蓝
0010	绿	1010	浅绿
0011	青	1011	浅青
0100	红	1100	浅红
0101	品红	1101	浅品红
0110	棕	1110	黄
0111	白	1111	强度白

显示器I/O

例：在屏幕上以红底蓝字显示字符串

```
data segment
string  db   'Hello world!'
len     equ   $-string
data ends
code segment
      assume cs:code, ds:data
start:
      mov  ax, data
      mov  ds, ax
```

```
mov  bp,seg string
mov  es,bp
mov   bp,offset string
mov   cx,len
Mov   dx,0
mov   bl,41h
mov   al,0
Mov   ah,13h
int      10h
Mov   ah,4ch
Int      21h
  code ends
end start
```

背景颜色组合

RGB	颜色
000	黑
001	蓝
010	绿
011	青
100	红
101	品红
110	棕
111	白

前景颜色组合

IRGB	颜色	IRGB	颜色
0000	黑	1000	灰
0001	蓝	1001	浅蓝
0010	绿	1010	浅绿
0011	青	1011	浅青
0100	红	1100	浅红
0101	品红	1101	浅品红
0110	棕	1110	黄
0111	白	1111	强度白

例9.17.asm

(8) 显示字符串（13号功能）

ES:BP=串地址

CX=串长度

DH,DL=起始行列

BH=页号

AL=0,BL=属性

串：char,char…char
光标返回到起始位置

AL=1,BL=属性

串：char,char…char
光标跟随移动

AL=2

串：char, attr, char, attr, …,
char, attr　光标返回起始位置

AL=3

串：char, attr, char, attr, …,
char, attr　　光标跟随移动

DOS显示功能调用

AH	功能	调用参数
2	显示一个字符（检CTRL_BREAK）	DL=字符；光标跟随移动
6	显示一个字符（不检CTRL_BREAK）	DL=字符；光标跟随移动
9	显示字符串	DS:DX=串地址 串必须以“$”结尾

wenjiecoder@outlook.com

显示器I/O

显示字符串 (09H功能)

入口参数：

定义要显示的字符串，字符串尾应为‘$’，作为结束显示的标志。

DS：DX = 字符串的首地址

功能号：AH = 09H

类型号：21H

出口参数：无

实现功能：显示字符串，遇‘$’停止显示，光标随动。

wenjiecoder@outlook.com

显示器I/O

例：编程显示字符串

```
data    SEGMENT                 ;定义显示的子符串
stri    DB  'Harbin Institute of Technology (Shenzhen)', '$'
data    ENDS
code    SEGMENT
        ASSUME  CS:code, DS:data
start:  MOV  AX, data           ;置缓冲区地址于DS:DX
        MOV     DS, AX
        LEA     DX, stri
        MOV     AH, 09H         ;调显示功能
        INT     21H
        MOV     AH, 4CH         ;返回DOS
        INT     21H
code    ENDS
        END     start
```


wenjiecoder@outlook.com

显示器I/O

例：利用DOS系统功能调用实现人机对话。根据屏幕上显示的提示信息，从键盘输入字符串并存入内存缓冲区。

```
DATA    SEGMENT
    BUF  DB  100              ;定义输入缓冲区长度
         DB    ?              ;保留为填入实际输入的字符个数
         DB    100 DUP(? )    ;准备接收键盘输入信息
    MESG  DB  'WHAT IS YOUR NAME ? $'      ;要显示的提示信息
DATA   ENDS
CODE    SEGMENT
        ASSUME  CS: CODE, DS: DATA
START: MOV    AX, DATA
       MOV    DS, AX
       ...
       MOV    DX, OFFSET MESG
       MOV    AH, 9                    ;屏幕显示提示信息
       INT    21H
       MOV    DX, OFFSET BUF
       MOV    AH, 10                   ;接收键盘输入
       INT    21H
       ...
```


wenjiecoder@outlook.com

BIOS与DOS简介

键盘I/O

显示器I/O

串行通信口I/O

文件存取I/O

wenjiecoder@outlook.com

串行通信口I/O

计算机与外设交换信息的过程中：

并行通信：多位数据通过多条数据线同时传送。

串行通信：多位数据通过同一条数据线按位传送。

并行通信就是把一个字符的各数位用几条线同时进行传输。与串行通信（一位一位传输）相比，在相同传输率下，并行通信的信息实际传输速度快、信息率高。

但并行通信比串行通信所用电缆多，随着距离的增加，电缆的开销会成为突出的问题。所以，并行通信总是用在数据传输率要求较高，而传输距离较短的场合。

wenjiecoder@outlook.com

按通信进行的过程，分为：单工、半双工、全双工通信方式

单　工：只容许数据由一方发、一方收，单向通讯。

2) 半双工：容许双向通讯，但是收发只能分时共用一路通道。

3) 全双工：容许数据同时双向收发。

串行通信可以分为两种类型：同步通信、异步通信。

异步通信

一个字符一个字符地传输，每个字符一位一位地传输，传输一个字符时，以起始位开始，然后传输字符本身的各位，接着传输校验位，最后以停止位
结束该字符的传输。一次传输的起始位、字符各位、校验位、停止位构成一组完整的信息，称为帧（Frame）。

帧与帧之间可有任意个空闲位。

起始位之后是数据的最低位。

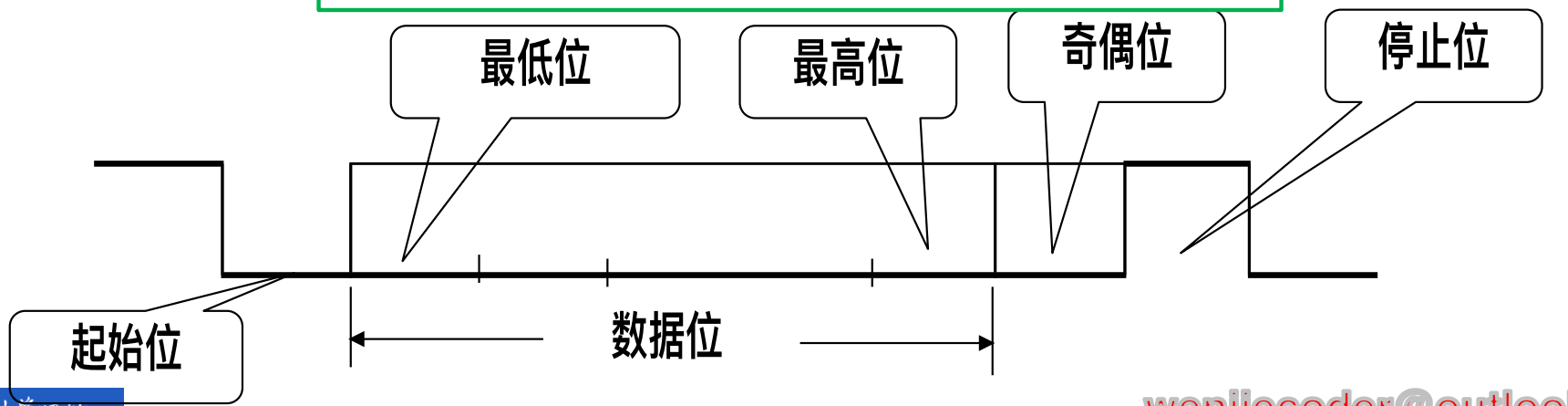
wenjiecoder@outlook.com

串行通信口I/O

异步通信：

- **起始位：**
  在数据发送线上规定无数据时电平为1，当要发送数据时，首先发送一个低电平0，表示数据传送的开始，这就是起始位。
- **数据位：**
  真正要传送的数据，由于字符编码方式不同，可以是5位、6位、7位、8位、9位等多位，数据位是由低位开始，高位结束（低位在前（左）、高位在后（右））；
- **奇偶校验：**
  数据发送完后，发送奇偶校验位，以检验数据传送的正确性，这种方法简单，容易实现。
- **停止位**：表示数据传送的结束，可以是1位、1.5位或2位。高电平1有效。

就数据传送而言，奇偶校验位是冗余位，但它表示数据的一种性质，这种性质用于检错，虽有限但很容易实现。



串行通信口I/O

奇偶校验

奇校验：原始码流+校验位 总共有奇数个1

偶校验：原始码流+校验位 总共有偶数个1

原始码	奇校验 奇数个1	偶校验 偶数个1
1011000	10110000	10110001
1010000	10100001	10100000
0011010	00110100	00110101

就数据传送而言，奇偶校验位是冗余位
，但它表示数据的一种性质，这种性质用于检错，虽有限但很容易实现。

wenjiecoder@outlook.com

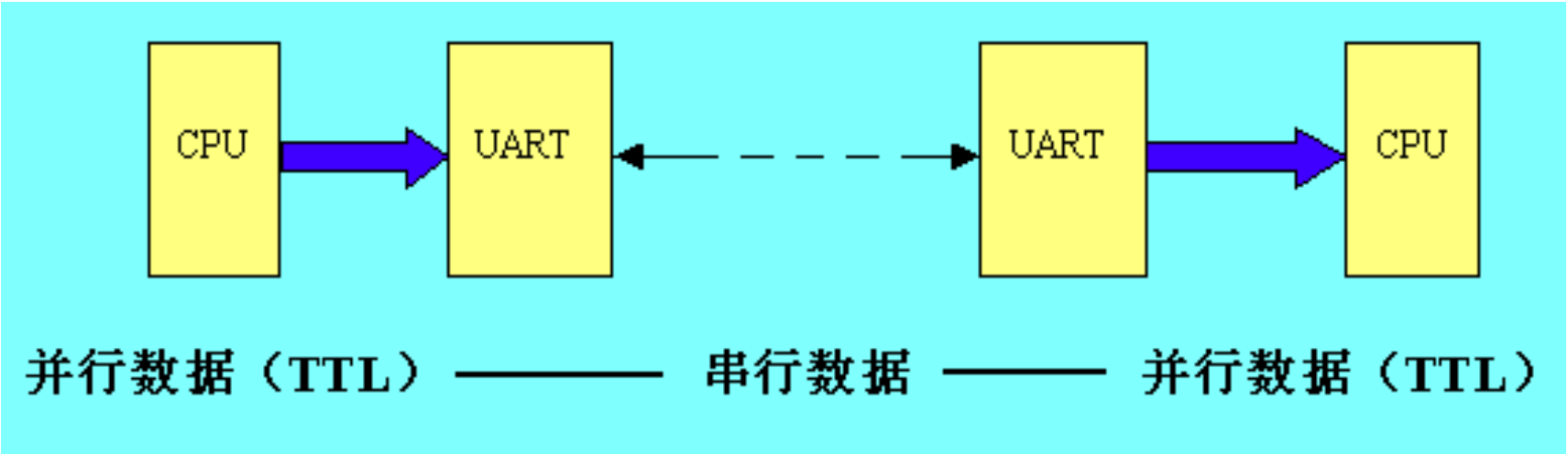
串行通信口I/O

专用IC芯片：通用异步接收/发送器（UNIVERSAL ASYNCHRONOUS RECEIVER /TRANSMITTER），UART。

作用：

1.数据的串行化(并行数据转化为串行数据)和反串行化（串行数据转化为并行数据）。

2.错误检验。

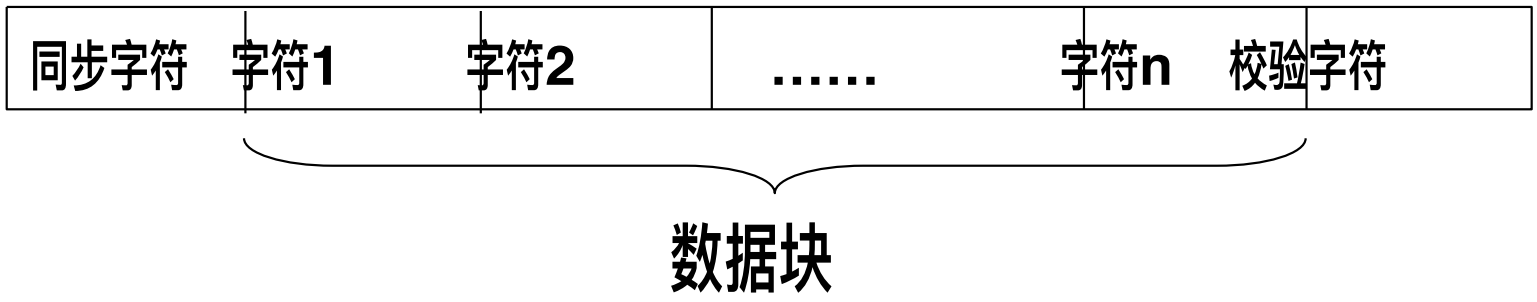


wenjiecoder@outlook.com

同步通信

- 在异步通信中，每个字符都要用起始位和结束位作为字符的开始和结束的标志，占用了传输的时间，所以在数据块传送时，为了提高效率，可以考虑去掉这些标志。

- 同步通信方式不给每个字符都加起始位和停止位，而把字符顺序的连接起来，组成一个数据块（首尾相连的数据串），把这样一个数据块称为一个信息桢。在数据块的开始加上一个同步字符，而在信息的末尾加有一定的差错检验字符，其格式如下：



串行通信口I/O

串行通信可以分为两种类型：同步通信、异步通信。

同步通信：

同步就是双方有一个共同的时钟，当发送时，接收方同时准备接收。
同步通信采用共用外部时钟来进行同步。

异步通信：

字符间同步，字符内部位与位之间也同步。

异步双方不需要共同的时钟，也就是接收方不知道发送方什么时候发送，所以在发送的信息中就要有提示接收方开始接收的信息，如开始位，结束时有停止位。
异步通信是按字符传输的。每传输一个字符就用起始位来进行收、发双方的同步。不会因收发双方的时钟频率的小的偏差导致错误。
这种传输方式利用每一帧的起、止信号来建立发送与接收之间的同步。
特点是：每帧内部各位均采用固定的时间间隔，而帧与帧之间的间隔时可以任意长。接收机完全靠每一帧的起始位和停止位来识别字符时正在进行传输还是传输结束。
字符间异步，字符内部各位同步。

串行通信口I/O

串行通信可以分为两种类型：同步通信、异步通信。

同步通信：

优点是可以实现高速度、大容量的数据传送；缺点是要求发生时钟和接收时钟保持严格同步，同时硬件复杂。

异步通信：

应用于在工业、实际应用中。适用于短距离、速率不高的情况下。

wenjiecoder@outlook.com

波特率和传输率

串行通信中，传输速率是用波特率来表示。所谓波特率是指单位时间内传送二进制数据的位数(简写为bps)。在计算机里，每秒传输多少位和波特率的含义是完全一致的。

收、发双方的波特率必须一致。

例：假设每个字符8位，1位起始位和1位终止位。计算串行传输5页，传输上述五页需要多少时间？（若数据传输率为9600bps）（每页80*25个字符）

解：每个字符10b

总数据量为：10*80*25*5=100000b

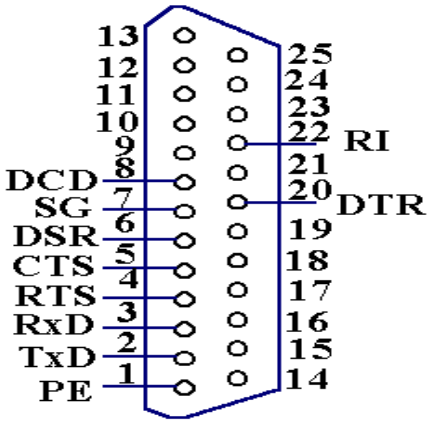
所需时间：100000/9600=10.4秒

wenjiecoder@outlook.com

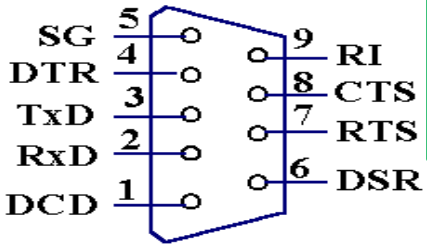
串行通信口I/O

RS232串行通信接口

- 在测控系统中，计算机通信主要采用异步串行通信方式，常用的异步串行通信接口标准有：
  - RS-232（RS-232A RS-232B RS-232C）
  - RS422 、RS485



DB-25



DB-9

RS-232C是一种常用的串行通信接口标准。
传统的RS-232C采用标准25芯D型插头座（DB25），后来使用简化为9芯D型插座（DB9）。

IBM PC通信端口

- IBM PC和兼容机可以连接4个通信端口，他们的编号为BIOS为：0-3，DOS为：1-4。
- 程序每次只能对其中一个端口进行存取。

DOS串行通信口功能调用

使用DOS命令可以设置串行通信参数，如波特率，校验位，字长和终止位。

常用的波特率：2400、4800、9600、19200、38400等。

格式：MODE COMm:b ,p ,d , s

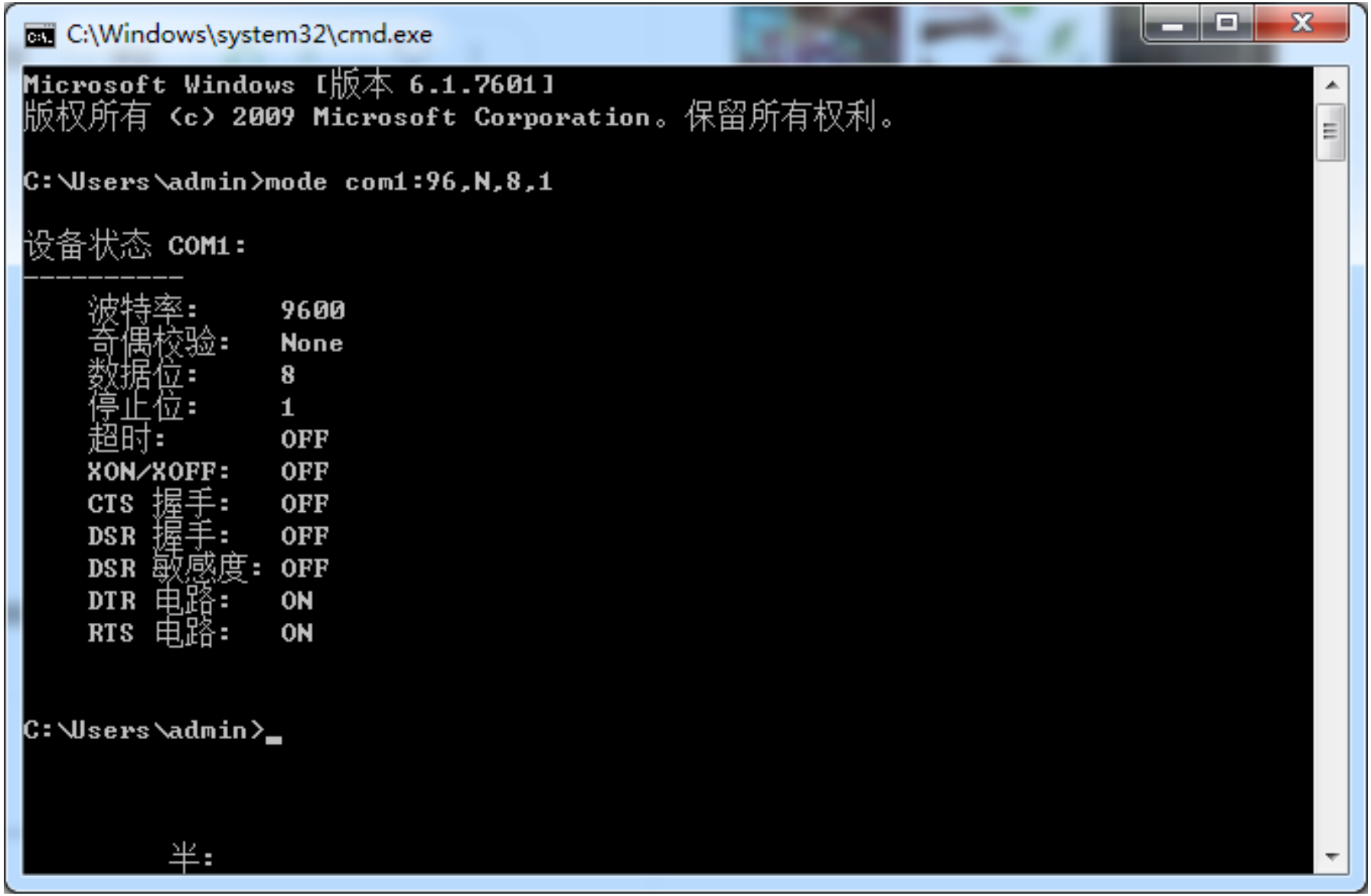
例如：MODE COM1:96,O,8,1

说明：m: 1~4, b：波特率，用波特率的最高两位来表示；

P：校验位（N：无校验，O：奇校验，E：偶校验）；d：数据的字长（5,6,7,8，默认值是7）；s：终止位位数（1，1.5，或2）。

wenjiecoder@outlook.com

串行通信口I/O



wenjiecoder@outlook.com

DOS串行通信口功能调用

AH	功能	调用参数	返回参数
3	从串行口读一个字符		AL=输入的数据
4	向串行口写一个字符	DL=输出的数据	

例：从串口读一个字符。

对通信端口COM1的功能调用

```
INPUT_CHAR DB  ?
…
MOV  AH, 3
INT   21H
MOV INPUT_CHAR, AL
…
```


wenjiecoder@outlook.com

串行通信口I/O

例：将字符串“HELLO”通过串口输出。

```
BUFFER  DB  ‘HELLO’
BUF_LEN  EQU  $-BUFFER
        …
        MOV  BX, OFFSET BUFFER
        MOV  CX, BUF_LEN
NEXT:
        MOV  DL, [BX]
        MOV  AH, 4
        INT    21H
        INC BX
        LOOP NEXT
        …
```


wenjiecoder@outlook.com

BIOS与DOS简介

键盘I/O

显示器I/O

串行通信口I/O

文件存取I/O

文件存取有关概念

文件存取功能调用

wenjiecoder@outlook.com

文件存取有关概念

- 文件操作既可以通过BIOS的中断服务INT 13H，也可以使用DOS系统功能调用INT 21H 。
- INT 13H提供的文件操作要求给出磁头号、磁道号、扇区号等磁盘物理参数，比较复杂。
- 而INT 21H提供的文件操作只要求给出文件名，相对要简单的多。
- 因此，本章主要介绍DOS系统功能调用21H的文件操作方法：

文件代号式磁盘存取。

相对文件全名，用文件代号指称文件效率更高，避免重复处理字符串

- 用户在处理一个文件时，必须给出完整的路径名，一旦文件的路径名送入操作系统，系统就返回给用户一个16位的二进制控制字，称为文件代号。
- 以后对该文件进行读写操作时，就用这个文件代号去查找相应的文件。

wenjiecoder@outlook.com

文件存取有关概念

1、路径名和ASCIZ串

ASCIZ串–

对文件进行说明。ASCIZ串最后一个字节为0，其余字节是指示文件位置的ASCII码字符串。

ASCII-ZERO

[d:][path]filename.exe, 00

其中d为驱动器名，path为路径名，.exe为文件名后缀。

用变量定义的形式就写作:

filename1 DB ‘C:\SAMPLE.TXT’,00

filename2 DB ‘c:\test\run.exe’, 00

文件存取有关概念

DOS已经预定义了文件代号0到4与标准输入输出设备对应，即

0 —— 标准输入设备，键盘；

1 —— 标准输出设备，屏幕；

2 —— 错误输出的标准设备，屏幕；

3 —— 标准辅助设备（通信端口）；

4 —— 标准打印设备。

在汇编语言或者操作系统看来，文件与标准输入输出设备都是数据流，两者的差别在于操作系统支持对文件的随机存取，而标准输入输出设备只能顺序存取。向标准输出设备写一段数据意味着把这些数据送到屏幕显示，从标准输入设备读一段数据则是从键盘读入一串数据。

这5个文件代号长期处于打开状态，应用程序可以直接使用。

对建立或打开的文件，其代号从5开始顺序排列，在任一时刻最多只能同时打开5个文件。当程序执行时，调用的每一个文件都必须分配一个唯一的文件代号。

wenjiecoder@outlook.com

文件存取有关概念

2、错误返回码

CF= 0，操作成功
CF= 1，操作失败

当用针对文件的DOS系统调用时，返回CF=0代表操作成功，返回CF=1代表操作失败。如果操作失败，会通过AX返回错误代码。

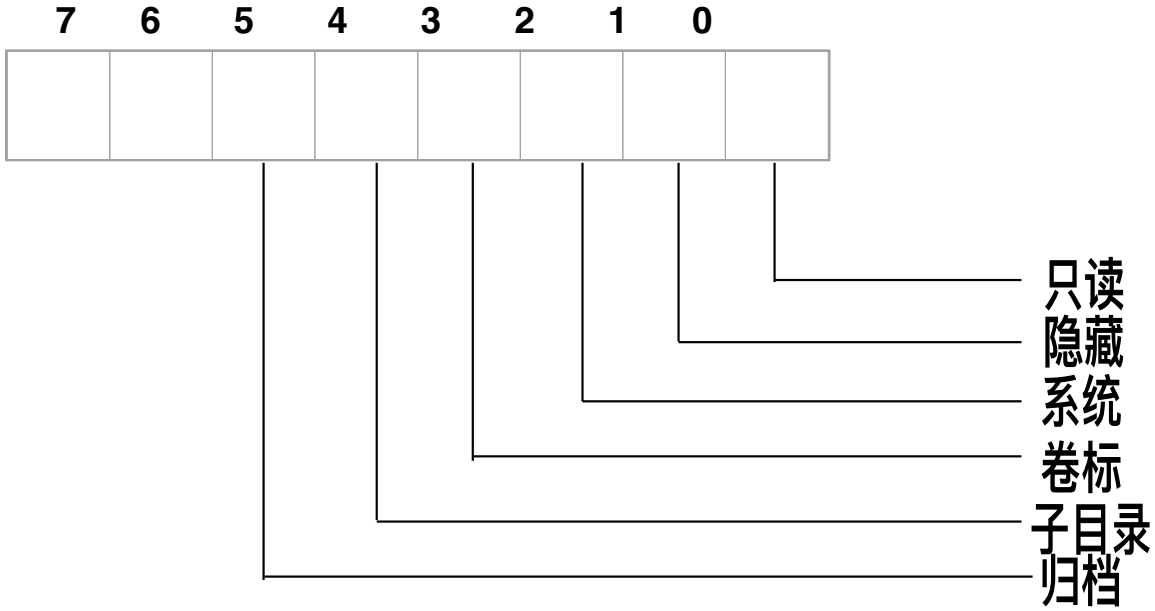
错误类型码保存在AX寄存器中。

部分错误代码的意义			
错误代码	意义	错误代码	意义
1	无效的功能号	10	不正确的环境
2	文件没有找到	11	格式无效
3	路径没有找到	12	访问方式无效
4	打开的文件太多	13	无效数据
5	拒绝存取	14	保留
6	无效的文件代号	15	无效盘符
7	内存控制快被破坏	16	试图删除当前目录
8	内存不足	17	设备不一致
9	内存快地址无效	18	已没有文件

wenjiecoder@outlook.com

文件存取有关概念

3、文件属性：每个文件都有自己的属性，用一个字节表示



一般情况下，用户文件只具有一种属性，如属性代码为**00H**的**普通文件**，属性代码为**01H**的**只读**文件，属性代码为**02H**的**隐藏文件**。重要的系统文件通常有多种属性，如属性代码为**07H**的文件，就具有**只读、隐藏、系统**三种属性。**属性字节**存放到**CX**寄存器中。

wenjiecoder@outlook.com

4、文件指针

- 使用DOS系统功能调用INT 21H
  建立文件或者打开文件成功后，DOS系统自动提供一个文件指针来指示文件的当前位置。
- 文件指针是一个32位二进制数，建立文件或者打开文件成功后，
  文件指针的初值为0，也就是指向文件的开始位置。
- 以后每次对文件的读写操作，系统自动修改文件指针的值，使文件指针指向下一次要读写的位置，
  每次文件指针的移动位移量就等于读写文件的字节数。

wenjiecoder@outlook.com

文件存取功能调用

常用扩展磁盘文件管理系统功能调用（21H）

AH	功能	入口参数	出口参数
3CH	建立文件	CX=文件属性 DS：DX=文件说明地址	CF=0，调用成功，AX=文件代号 CF=1，调用失败，AX=错误代码
3DH	打开文件	AL=存取代码 DS：DX=文件说明地址	CF=0，调用成功，AX=文件代号 CF=1，调用失败，AX=错误代码
3EH	关闭文件	BX=文件代号	CF=0，调用成功 CF=1，调用失败，AX=错误代码
3FH	读文件	BX=文件代号 CX=读文件的字节数 DS：DX=文件缓冲区地址	CF=0，调用成功，AX=实际读入的字节数 CF=1，调用失败，AX=错误代码
40H	写文件	BX=文件代号 CX=写文件的字节数 DS：DX=文件缓冲区地址	CF=0，调用成功，AX=实际写入的字节数 CF=1，调用失败，AX=错误代码
42H	移动文件指针	BX=文件代号 AL=移动方式 CX：DX=移动字节数	CF=0，调用成功，DX：AX=指针新位置 CF=1，调用失败，AX=错误代码
43H	读写文件属性	AL=0检验/1置文件属性 CX=文件属性 DS：DX=文件说明地址	CF=0，调用成功，AX=文件属性 CF=1，调用失败，AX=错误代码

文件存取功能调用

文件存取代码：

位	存取代码	位	存取代码
0-2	000=为读而打开文件	3	1=保留
	001=为写而打开文件	4-6	共享方式
	002=为读和写而打开文件	7	继承标志

wenjiecoder@outlook.com

文件存取功能调用

一、建立磁盘文件

建立一个新文件或用同一个文件名重写一个旧文件时，**首先要建立文件并赋给它一个属性**，如果DOS发现要建立的文件已经存在，那么原来的文件就被破坏。

例：建立一个有正常属性文件的指令序列

```
PATHNM1 DB ‘E:\ACCOUNTS.FIL’,00H
HANDLE1 DW  ?
…
MOV AH,3CH
MOV CX,00
LEA DX, PATHNM1
INT 21H
JC  ERROR
MOV HANDLE1, AX
…
```

文件代号

建立文件（3CH）

功能：按指定文件名建立文件。

入口参数：

(AH)=3CH,

DS:DX指向ASCIZ字符串的段地址和偏移地址

(CX)=文件属性。

出口参数：

若成功，则CF=0，(AX)=文件代号；

若失败，则CF=1，(AX)=错误代码。

文件存取功能调用

二、打开和关闭磁盘文件

```
PATH DB ‘E:\ACCOUNTS.FIL’,00H
HANDLE DW  ?
…
;Open the file
MOV  AH, 3DH
MOV  AL, 0
LEA   DX, PATH
INT    21H
JC      ERROR
MOV HANDLE, AX
…
;Close the file
MOV AH,3EH
MOV BX, HANDLE
INT    21H
JC      ERROR
…
```

打开文件（3DH）

功能：打开由ASCII Z串指定的文件。

入口参数：

(AH)=3DH，DS：DX指向ASCIZ字符串的段地址和偏移地址，(AL)=存取代码（0：读文件，1：写文件，2：读、写文件）。

出口参数：

若成功，则CF=0，（AX）=文件代号；

若失败，则CF=1，（AX）=错误代码。

关闭文件（3EH）

功能：关闭文件代号指定的文件。

入口参数：

(AH)=3EH，(BX)=文件代号。

出口参数：

若操作成功，则CF=0；

若操作失败，则CF=1，（AX）=错误代码。

文件存取功能调用

三、读磁盘文件

例：从文件readme.txt中读512字节到缓冲区。

```
PATH DB ‘E:\readme.txt’,00H
HANDLE DW  ?
Data   DB  512  DUP(?)
…
;Open the file
MOV  AH, 3DH
MOV  AL, 0
LEA   DX, PATH
INT    21H
JC     ERROR
MOV HANDLE, AX
…
;Read the file
MOV AH,3FH
MOV BX, HANDLE
MOV CX, 512
LEA   DX, data
INT    21H
JC     ERROR
CMP  Ax, 0
JE     EndFile   ;表示文件为空
…
```

读文件（3FH）

入口参数：

(AH)＝3FH，(BX)＝文件代号，(CX)＝要读取的字节数；

DS：DX指向接收数据缓冲区的段地址和偏移地址。

出口参数：

若成功，则CF＝0，(AX)＝实际读入字节数，(AX)＝0，文件结束；

若失败，则CF＝1，(AX)＝错误代码。

当文件读入操作完成后，要及时关闭。

文件存取功能调用

四、写磁盘文件

例：把OUTREC数据区中的256个字节写入磁盘文件

```
HENDLE DW ?
OUTREC DB 256 DUP(?)
…
MOV AH,40H
MOV BX, HANDLE
MOV CX, 256
LEA DX, OUTREC
INT 21H
JC ERROR1
…
;Close the file
MOV AH,3EH
MOV BX, HANDLE
INT   21H
JC    ERROR2
…
```

写文件（40H）

入口参数：

(AH)=40H，(BX)=文件代号，(CX)=要写入的字节数；

DS：DX指向存放写入信息数据缓冲区的段地址和偏移地址。

出口参数：

若成功，则CF=0，(AX)=写入字节数；

若失败，则CF=1，(AX)=错误代码。

当文件写入操作完成后，必须用DOS功能调用3EH来关闭文件，以确保操作系统将文件记录在磁盘上。

文件存取功能调用

例：从文件file1中读取10个字符到file2文件中。

```
data segment
    fname db 'c:\file1.dat',00
    fname2 db 'c:\file2.dat',00
    dta  db 80h dup(0)
data ends
code segment
    assume cs:code,ds:data
start:mov ax,data
     mov ds,ax
     mov es,ax
     mov dx,offset fname  ;Open source file
     mov al,0             ;read file
     mov ah,3dh           ;Open the file
     int 21h
     mov si,ax            ;file number
     mov bx,si
     mov dx,offset dta    ;read from the source file
     mov cx,10            ;read 10 bytes
     mov ah,3fh           ;read the file
     int 21h
     mov di,ax            ;实际读入的字节数
     mov ah,3eh           ;close the source file
     int 21h
     mov dx,offset fname2 ;make the new file
     mov cx,0
     mov ah,3ch
     int 21h
     mov si,ax            ; 文件代码
     mov dx,offset dta    ;write into the file
     mov cx,di            ;write bytes
     mov bx,si            ;file number
     mov ah,40h           ;write
     int 21h
     mov bx,si            ;close the file
     mov ah,3eh
     int 21h
     mov ah,4ch
     int 21h
code ends
end start
```

文件存取代号

文件代号

wenjiecoder@outlook.com

文件存取功能调用

扩展：有关文件外部特性与目录的操作

子功能号(AH)	功 能	入口参数	出口参数
39H	建立子目录	DS:DX=路径字符串首地址	
3AH	删除子目录	DS:DX=路径字符串首地址	
3BH	改变当前目录	DS:DX=路径字符串首地址	
41H	删除文件	DS:DX=待删除文件名字符串首地址	
43H	置/取文件属性	DS:DX=文件名字符串首地址 AL=0取文件属性 AL=1置文件属性 CX=文件属性	取文件属性成功时，CX=文件属性
47H	取当前目录路径	DL=驱动器号 DS:SI=65字节的数据缓冲区	成功时，缓冲区中被填写当前目录(含路径)字符串
57H	置/取文件日期和时间	BX=文件代号 AL=0取文件的日期和时间 AL=1置文件的日期和时间 CX=时间，DX=日期	取日期和时间成功时，CX=时间，DX=日期

作业：

P354：

9.3、9.4

P433：

11.4、11.5

wenjiecoder@outlook.com

复习与考试

基本概念要清楚

课堂中讲过的例题要吃透

常用的指令及其使用方式要记牢

常用的BIOS和DOS系统功能调用要掌握

作业和实验中出现的典型问题要重视

wenjiecoder@outlook.com

复习与考试

期末考试题型（初步计划）：

选择题

填空题

简答题

分析题

编程题

概念清楚

思路严谨

分析全面

答题完整

课程成绩计算方式：

作业成绩占15%

实验成绩占25%

期末成绩占60%

wenjiecoder@outlook.com

预祝大家取得好成绩！

谢谢！

wenjiecoder@outlook.com