第四章 中断调用指令

INT 21H

INT 10H

宏的概念

中断：

引发中断有两种可能：

1. 硬件引发
2. 软件引发

通过INT（interrupt）指令或者异常来产生。

没有中断的时候，只能对外围设备进行轮询。

有了中断，外围设备可以向CPU请求响应中断。

中断用字节来表示，中断最多可以有256个，从00H~FFH

响应中断：去执行已知对应的中断服务程序。

中断服务程序要存储在内存中。

中断请求的操作过程：

1. 执行中断程序的时候也有保存现场，不仅是断点（IP）的保护，还有对标志寄存器的保护。
2. 找到相应的中断服务程序，并执行

把所有的中断服务程序放到一起，然后把每个中断服务程序的入口地址放到一个表中，这个表就是中断向量表。

根据这个表去查找到相应的中断服务程序的入口地址。

入口地址：段地址 + 偏移地址

两个地址都是16位的，每个入口地址都占用32位，即4个字节。

整个中断向量表占用的空间为：4 × 256 = 1024字节 = 1KB

内存的最开始1K用来存储中断向量表。

中断向量表是按中断号顺序存放的。

比如0000 ~ 0003就是第一号中断……

只要把中断号乘以4就是它对应的中断在中断向量表的地址。

中断和子程序调用的区别：

中断向量号一旦确定，它的中断服务程序在内存中的位置是固定的，而子程序调用时，子程序在内存中的位置是随机的。

中断的发生是随机的，子程序调用在程序写出来之后是确定的。

有些中断请求是可以屏蔽掉的，而子程序调用是不能屏蔽的。

返回指令：中断服务返回是IRET，要把FR也恢复出来，子程序调用RET，不会改变FR的值。

硬件中断：INTR和NMI这两个引脚用来接收中断请求，高电平就是接收到中断请求。

NMI获得的中断是不可屏蔽中断，INTR接收的中断请求是可屏蔽中断，只要IF = 0就是屏蔽中断请求。

软件中断：由INT指令引发的中断，与外围设备没有关系的。

00H~1FH这32个中断是被BIOS占用的。

20H~3FH这32个中断是被DOS占用的。

中断服务调用的时候要传递一个参数，汇编语言中，参数的传递主要是通过寄存器来传递，也可以通过内存（堆栈、数据段）来传递。

INT 10H BIOS中断

已经固化在BIOS中。

实现了汇编语言程序员与显示器之间的沟通。

这里通过寄存器AH来传递参数。

功能：

1. 清屏

AH = 06（）或者AH = 07

AL = 00：整个页面都清掉

BH = 07：正常属性

CH = 00：起点（左上角）的行数

CL = 00：起点的列数

DH = 24：终点（右下角）的行数

DL = 79：终点的列数

1. 设置光标位置

AH = 02：设置光标

BH = 00：设置到第0页

DH = 20：光标的行的坐标

DL = 20：光标的列的坐标

AH = 03：获取当前光标的位置

BH = 00：第0页。

DH和DL分别返回了光标所在的行和列

CX返回了当前光标的形状。

1. 显示一个字符
2. 画线画图
3. 改变显示模式

传递一个ASCII码时，传递一个字节表示ASCII码，还有一个字节用来表示显示的模式。

1. 文本模式

25行80列字符。

1. 视频模式

AH = 00

AL = 视频模式

INT 21H DOS中断

功能：

1. 在显示器中显示一个字符串

AH = 09：显示字符串

DX = 字符串起始地址。

默认碰到第一个’$’时，结束输出，且’$’不会被输出。

AH = 02：显示一个字符

DL = 要显示的字符

1. 从键盘读入一个字符

AH = 01：接收从键盘输入的一个字符

输入的字符存放在AL中。

1. 从键盘读入一个字符串

AH = 0AH：读入一个字符串

DX = 字符串存放的首地址

默认接收的时候，回车结束输入，并且回车会被读入

DB 6, ?, 6DUP(0FFH)

第一个单元定义了最大接收6个字符（包含回车），其中输入了多少个字符会存放到第二个单元（就是那个?）中，从第三个单元开始才是真正的字符串。

如果输入超过了字符串的大小，则只会读取前面的字符，加上最后一个回车符。

1. 不带回显的输入

AH = 07

回车与换行：

回车（CR）：0DH

光标移动到这一行的首部

换行（LF）：0AH

光标向下移动一行

LABEL指令

可以使某个变量或者函数有多个名称。

name LABEL attribute

attribute可以是BYTE、WORD、DWORD、QWORD、DT

DATA1 LABEL BYTE

MOV AL, DATA1

每次读取一个字节，如果BYTE改成WORD，则每次读取一个字。

如上面的字符串：

DATA1 DW 6, ?, 6DUP(?)

DATA1+2才是字符串的第一个字符，可以用：

BUF\_COUNT

宏（macro）

子程序：把一些经常需要执行的功能和操作放到子程序中，而不是每次都去把这个功能写一遍。

宏：把一些经常需要用到的操作或者功能写到一个宏中，需要用到的时候去把宏扩展出来。把反复出现的指令写到宏中。

区别：

子程序在内存中只在一个地方存储，每一次调用子程序的时候都会到同一个地方执行。

宏扩展是把程序变长了，有多少次宏的扩展就要占用多少内存，而不是每次都到同一个地方去。

宏是在汇编的时候已经把宏扩展嵌入到代码里面进去了，不会在内存中存有单独的宏。

1. 子程序调用在内存中占用的空间比宏扩展少
2. 执行过程中，宏比子程序调用要快

宏的定义：

name MACRO 参数列表（寄存器、内存单元、立即数都可以）

……（具体指令）

ENDM

如移位的时候需要把移位次数放到CL中，怕忘记就写成一个宏，然后把移位次数传进去就可以了。

宏定义里面的指令如果出现标签，要注意标签的唯一性，宏多次调用可能会产生不唯一，要进行处理。

第五章 键盘和鼠标中断

键盘中断

INT 16H

用一个状态字节来表示CTRL、shift、ALT键是否被按下，来判断按下的字母的大小写等。

鼠标中断

INT 33H

通过AX来传递参数。