3.12 中断调用与返回指令

- 中断处理过程

  计算机在程序运行过程中，由于发生了某些“紧急事件”，需要进行特殊处理（转向中断服务子程序执行），处理后返回到程序中断处继续执行，这种过程称为中断处理过程。

- “紧急事件”可以是由硬件产生的，例如系统掉电、硬件故障、定时计数器溢出等；可以是执行指令产生的，例如除法溢出等；还可以是专门的中断调用指令所产生的。
- 本节主要介绍专门的中断调用指令、中断服务子程序的结构和中断返回指令。

3.12 中断调用与返回指令

3.12.1 中断调用指令INT

中断调用指令INT的格式为：

INT　n

表示调用第n号中断，n为中断类型号，其值可以是0～255。

当执行该指令时，会进行下列操作：

- PSW、CS、IP入栈；
- 清除IF、TF标志；
- 从中断向量表中取出中断向量（中断向量为中断服务子程序的入口地址）；
- 转到中断服务子程序进行执行；
- 中断服务子程序的最后一条指令应该是中断返回指令，执行该指令可以返回到程序中断处继续执行。

3.12 中断调用与返回指令

3.12.2 中断返回指令IRET

中断返回指令IRET的格式为：

IRET

表示从中断服务子程序返回主程序。

执行该指令时，会完成IP、CS、PSW的出栈操作，其次序与INT n 指令的入栈次序相反。由于修改了IP和CS的内容，从而实现了程序的返回功能。

3.12 中断调用与返回指令

3.12.3中断向量表

- 每个中断服务子程序的入口地址（称为中断向量）为32位（16位的偏移地址和16位的段地址），占用4个地址单元；
- 计算机中采用最低的1024个地址单元（称为0页）来存储中断向量，因此，n号中断的中断向量存放地址为：4×n，调用该中断时，可以从该地址获得中断服务子程序的入口地址。

3.12 中断调用与返回指令

3.12.4 中断服务子程序结构

- 由于中断是随时可能调用的程序，所以无法预知其调用的位置，这样在设计中断服务子程序时，应该确保所有寄存器的内容保持不变。
- 在中断服务子程序的入口处，需要将用到的寄存器通过堆栈进行保护，在中断程序的返回之前，从堆栈中恢复寄存器的内容。
- 中断服务子程序的结构：

3.12 中断调用与返回指令

<中断子程序名>:

```
        PUSH   AX     ；保护现场
          -
          -
          -
        PUSH  SI
          -
          -           ；中断子程序主体
          -
        POP    SI     ；恢复现场
          -
          -
          -
         POP    AX
         IRET          ；中断返回
```

3.12 中断调用与返回指令

3.12.5 系统功能调用

IBM PC/XT机器提供了许多中断服务子程序，在编写程序时可以直接进行调用。

系统功能调用分两类：

- BIOS功能调用，它是以系统中所支持的硬件为对象的，例如13H号功能为磁盘服务，17H号功能为打印机服务；
- DOS 功能调用，它以软件方式支持的功能，例如常用的DOS功能调用有21H、20H、25H、26H等，详细说明参见附录。

3.12 中断调用与返回指令

- 单字符输入（带回显）

  21H中断，第1号功能（AH=01H），（AL）=读取字符的ASCII

  程序片段：

```
MOV  AH, 01H
INT   21H
```

- 字符串输入

  21H中断，第10（0A）号功能（AH=0AH），（DX）=串首址

  例：数据段定义

```
KEYBUFF   DB   100
          DB    ?
          DB   100 DUP(?)
```

  程序片段：

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

3.12 中断调用与返回指令

- 单字符输出

21H中断，第2号功能，（DL）=待显示字符的ASCII

程序片段：

```
MOV   AH, 02H
MOV   DL, ‘X’
INT    21H
```

- 字符串输出

21H中断，第9号功能（AH=9），（DX）=字符串首址

例： 数据段定义

```
TABLE  DB  ‘PLEASE INPUT YOUR NAME’,0DH,0AH,’$’
```

程序片段：

```
MOV  AH, 09H
LEA   DX, TABLE
INT    21H
```