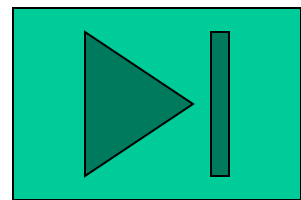


第十五章 键盘程序设计

15.1 键盘的种类与扫描码

15.2 IBM PC/XT标准键盘接口

15.3 键盘中断程序设计



15.1 键盘的种类与扫描码

| | | | | | | | | | | | | | | | | | | | | | |
|----------|-----------|-------------|----------|----------------|----------------|-----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|------------------|-----------------|----------------|--------------|-------------------|--------------|------------------------|
| 59 F1 | 60 F2 | 1 ESC | 2 ! 1 | 3 @ 2 | 4 # 3 | 5 \$ 4 | 6 % 5 | 7 ^ 6 | 8 & 7 | 9 * 8 | 10 (9 | 11) 0 | 12 _ - | 13 + = | 14 ←Backspace | | 69 Num Lock | | 78 Scroll Lock | | |
| 61 F3 | 62 F4 | 15 ← → | | 16 Q | 17 W | 18 E | 19 R | 20 T | 21 Y | 22 U | 23 I | 24 O | 25 P | 26 { [| 27 }] | 28 ENTER | | 71 7 Home | 72 8 ↑PgUp | 73 9 PgDn | 74 - 78 * |
| 63 F5 | 64 F6 | 29 Ctrl | | 30 A | 31 S | 32 D | 33 F | 34 G | 35 H | 36 J | 37 K | 38 L | 39 : ; | 40 “ , | 41 ~ ' | | | 75 4 ← | 76 5 | 77 6 → | |
| 65 F7 | 66 F8 | 42 Shift | | 43 \ Z | 44 Z | 45 X | 46 C | 47 V | 48 B | 49 N | 50 M | 51 < , | 52 > . | 53 ? / | 54 Shift | 55 * prtsc | 79 1 End | | 80 2 ↓PgDn | 82 | 28 Enter |
| 67 F9 | 68 F10 | 56 Alt | | | 57 Spacebar | | | | | | | | | | | 58 Caps Lock | | 82 8 Ins | 83 Del | | |

83键的键位布局和扫描码

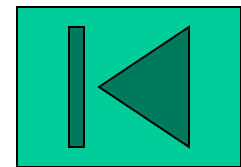
15.1 键盘的种类与扫描码

| | | | | | | | | | | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-------------------|-------|---|--|--|
| 1 ESC | 59 F1 | 60 F2 | 61 F3 | 62 F4 | 63 F5 | 64 F6 | 65 F7 | 66 F8 | 67 F9 | 68 F10 | 69 F11 | 70 F12 | Prt Sc | 70 SRC Lock | Pause | Num Lock <input type="checkbox"/> | Caps Lock <input type="checkbox"/> | Scroll Lock <input type="checkbox"/> |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-------------------|-------|---|--|--|

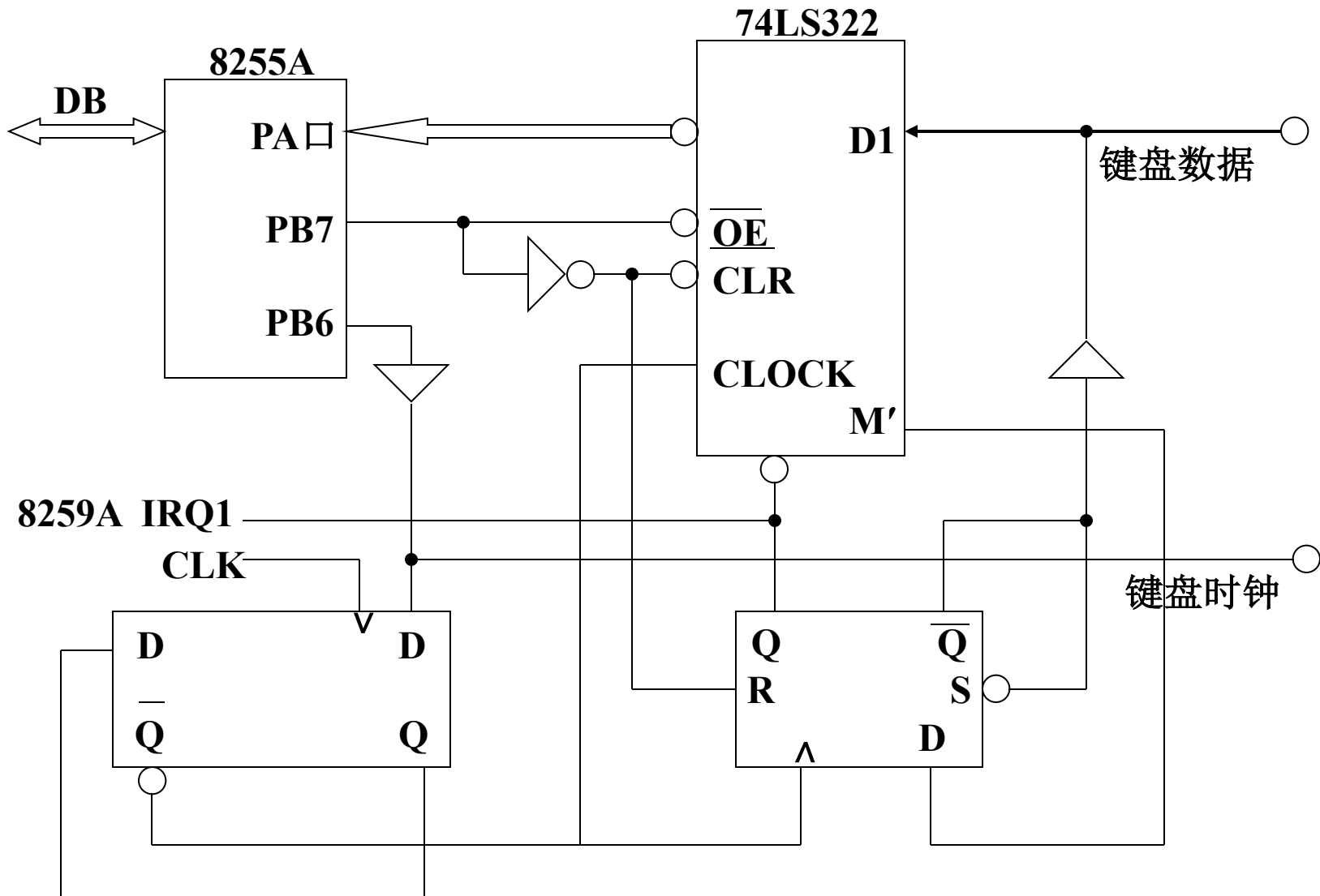
| | | | | | | | | | | | | | | | | | |
|----------------|-----------|----------|----------------|-----------|----------|----------|----------|-----------|-----------|-----------|-------------|---------------|------------------|--|--|--|--|
| 41 ~ ,` | 2 ! 1 | 3 @ 2 | 4 # 3 | 5 \$ 4 | 6 % 5 | 7 ^ 6 | 8 & 7 | 9 * 8 | 10 (9 | 11) 0 | 12 _ - | 13 + = | 14 ←Backspace | | | | |
| 15 ← → | 16 Q | 17 W | 18 E | 19 R | 20 T | 21 Y | 22 U | 23 I | 24 O | 25 P | 26 { [| 27 }] | 28 | | | | |
| 29 Capslock | 30 A | 31 S | 32 D | 33 F | 34 G | 35 H | 36 J | 37 K | 38 L | 39 : ; | 40 " ' | ENTER | | | | | |
| 42 Shift | 44 Z | 45 X | 46 C | 47 V | 48 B | 49 N | 50 M | 51 < , | 52 > . | 53 ? / | 57 Shift | 43 \ | | | | | |
| 29 Ctrl | 56 Alt | | 57 Spacebar | | | | | | | | Alt | | Ctrl | | | | |

| | | | | | | |
|-----|------|--------------|---------------|-----------|--------------|---------|
| | | Page Up | 69Num Lock | | | 74 - |
| Ins | Home | Page Down | 71 7 Home | 72 8 ↑ | 73 9 PgUp | 78 * |
| Del | End | | 75 4 ← | 76 5 → | 77 6 | |
| | | | 79 1 End | 80 2 ↓ | 81 3 PgDn | |
| | | | 82 Ins | | 83 Del | |

101键的键位布局 and 扫描码



15.2 IBM PC/XT标准键盘接口



15.2 IBM PC/XT标准键盘接口

1. 信号连接及使用

(1) 系统加电后，将8255A初始化为方式0，PA口为输入方式，PB口为输出方式。

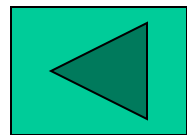
(2) PB口的PB6用于控制键盘时钟：

当PB6=0时，禁止键盘输出；当PB6=1时，允许键盘输出。

(3) PB口的PB7控制74LS322的输出：

PB7=0时，允许74LS322输出；PB7=1时，禁止74LS322输出。同时，PB7还用于清除74LS322和中断请求触发器。

(4) 中断请求触发器的Q接至8259A的IRQ1端， \bar{Q} 接至键盘数据输入端，以保证在键盘数据未取出之前，禁止74LS322接收新数据。



15.2 IBM PC/XT标准键盘接口

2. 工作过程

(1) 将8255A的PB6置为1，PB7置为0，键盘逻辑可正常工作。

(2) 当用户按下某一键时，键位的扫描码的串行数据通过电缆送至74LS322的D1端。

(3) 74LS322按串行数据格式接收完一个扫描码后，保存起来，串行数据的起始位通过M'端送入触发器，使触发器置1，产生中断请求信号，同时，封锁键盘数据线。

(4) CPU响应中断，从8255A中读取扫描码，然后通过PB7发出一个正脉冲信号，清除74LS322寄存器和中断请求触发器，以备接收下一个扫描码。

键按下和键抬起各产生一次中断请求，由扫描码的最高位识别，最高位为0，表示键按下，最高位为1，表示键抬起。

(5) 将读取的键位扫描码译为ASCII码，然后按需要进行解释。

15.3 键盘中断程序设计

例15.1 编制从键盘上接收字符并在屏幕上显示出来的程序，按ESC键则返回系统。

从键盘上接收字符利用键盘中断服务程序实现，每当按下一键时，就进入中断服务程序，中断服务程序读取按键的扫描码，并进行译码，放入键盘缓冲区。

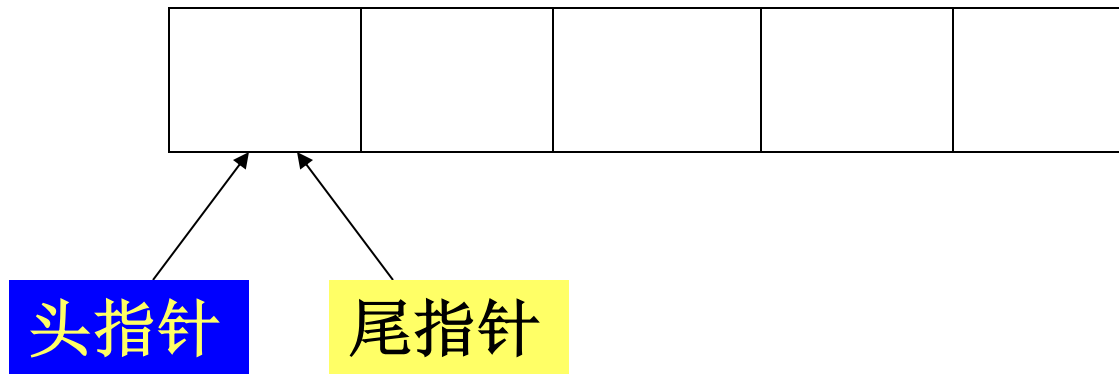
在屏幕上显示字符，利用DOS中断的2号功能实现。主程序从键盘缓冲区中读取字符，利用DOS中断的2号功能在屏幕上显示出来。

由此可见，键盘中断服务程序和主程序通过键盘缓冲区交换信息。键盘中断服务程序向缓冲区中写信息，而主程序则从键盘缓冲区中读取信息。

15.3 键盘中断程序设计

键盘缓冲区是一个循环队列，按照先进先出的原则存取信息，当访问到最后一个单元时，再从头开始继续访问。示意如下：

(1) 初始化后

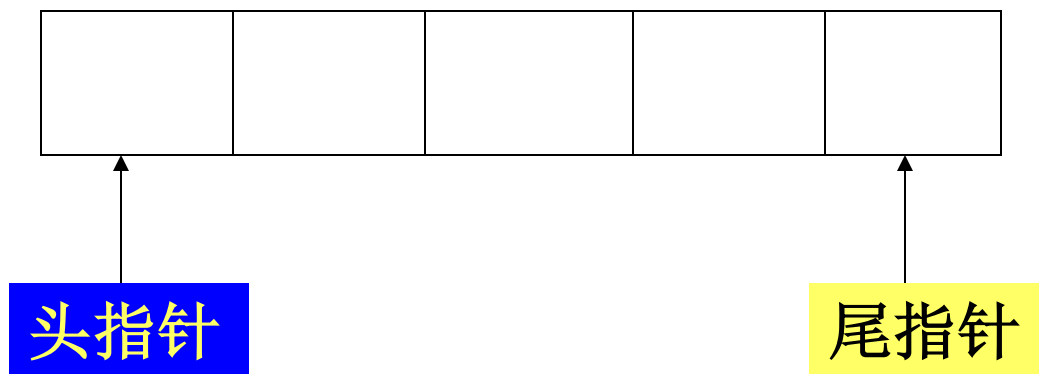


头指针：用于指示要读的信息在缓冲区中的位置

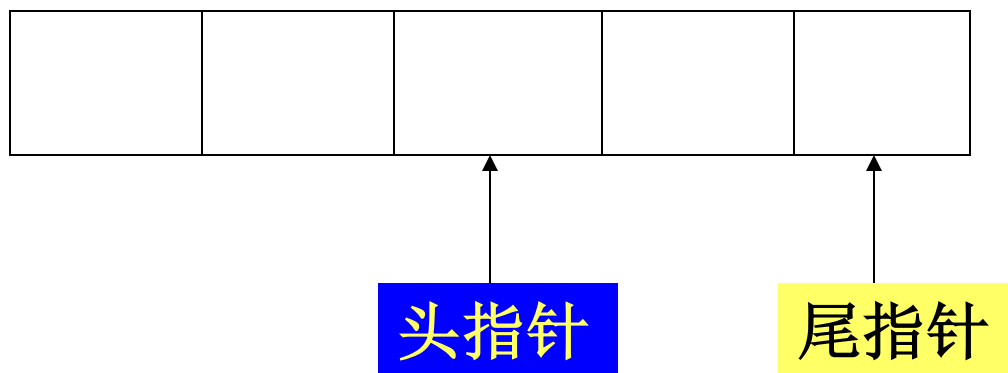
尾指针：用于指示要写入的单元的位置

15.3 键盘中断程序设计

(2) 写入4个数据后

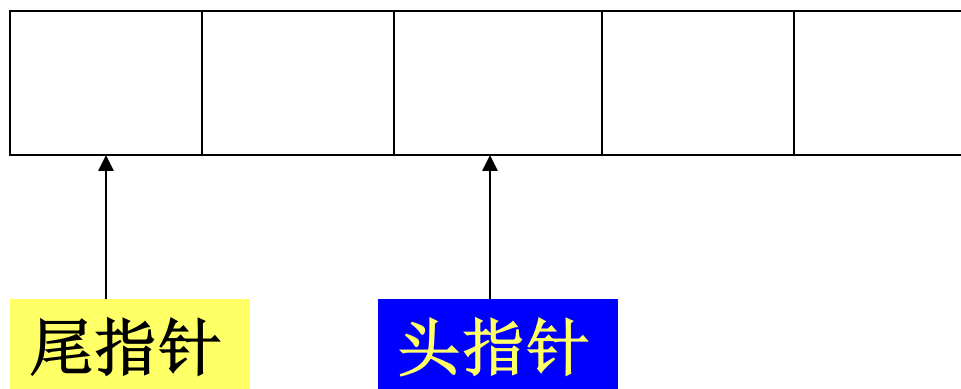


(3) 读取2个数据后

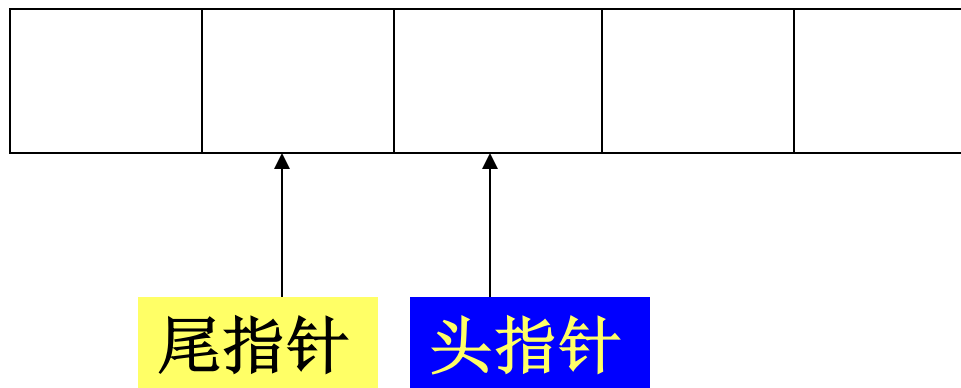


15.3 键盘中断程序设计

(4) 写入1个数据后

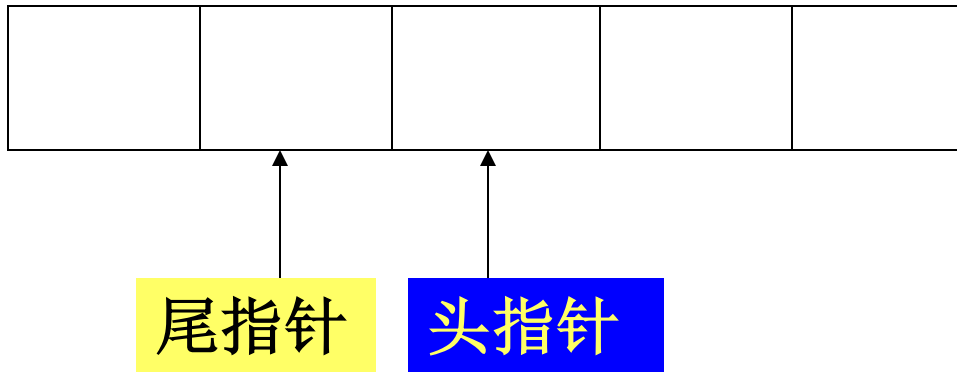


(5) 再写入1个数据后



15.3 键盘中断程序设计

(6) 此时再写入1个数据



缓冲区满状态:

$(\text{尾指针} + 2) \text{ MOD 缓冲区的长度} = \text{头指针}$

缓冲区空状态:

头指针 = 尾指针

| | | |
|--------------|----------------|--|
| SSEG | SEGMENT | STACK |
| | DB | 80 DUP (0) |
| SSEG | ENDS | |
| DATA | SEGMENT | |
| KEYIP | DW | ? ; 保存原键盘服务IP单元 |
| KEYCS | DW | ? ; 保存原键盘服务CS单元 |
| BTOP | DW | ? ; 缓冲区头指针 |
| BTLL | DW | ? ; 缓冲区尾指针 |
| BUFF | DW | 16 DUP (0) ; 缓冲区 |
| SCTAB | DB | 0, 1BH, '1234567890-=', 08H ; 转换表 |
| | DB | 09H, 'QWERTYUIOP[]', 0DH |
| | DB | 0, 'ASDFGHJKL;'' |
| | DB | 0, '\ZXCVCBNM,./', 0 |
| | DB | 0, 0, 20H, 13 DUP (0) |
| | DB | '789-456+1230.', 0 |
| DATA | ENDS | |

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,SS:SSEG

KBPRG PROC FAR

PUSH DS

XOR AX,AX

PUSH AX

MOV BX,DATA

MOV DS,BX

CLI ; 清中断

CLD ; 关中断

MOV ES,AX ; 0段

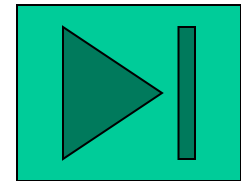
MOV DI,9*4 ; 键盘中断向量

MOV AX,ES:[DI] ; 取原键盘服务IP

MOV KEYIP,AX ; 保存IP

LEA AX,KBINT ; 取新键盘IP

STOSW ; 送入中断向量表



MOV **KEYCS,AX**

主缓冲区

指针

指针

键盘申请中断

斤

中区头指针

指针比较

空，跳

键盘扫描码和ASCII码

| | | | |
|--------------|------------|--------------------------|---------------|
| | ADD | BX,2 | ； 缓冲区头指针+2 |
| | CMP | BX,OFFSET BUFF+32 | ； 超出缓冲区了？ |
| | JNE | DISP | ； 未超跳转 |
| | LEA | BX,BUFF | ； 已超，循环至缓冲区始 |
| DISP: | MOV | BTOP,BX | ； 保存新的头指针 |
| | STI | | |
| | CMP | AL,0 | ； 是略去的键值？ |
| | JZ | NEXT | ； 是，去接受新的值 |
| | CMP | AL,1BH | ； 是ESC吗？ |
| | JE | EXIT | ； 是则，返回操作系统 |
| | MOV | AH,2 | ； 显示键入的字符 |
| | MOV | DL,AL | |
| | INT | 21H | |
| | CMP | DL,0DH | ； 是回车符？ |
| | JNZ | NEXT | ； 不是，则去接收新的字符 |
| | MOV | AH,2 | ； 是则，输出换行符 |

```
        MOV     DL,0AH
        INT     21H
        JMP     NEXT          ; 等待接收新的键值
EXIT:   CLI
        SUB     DI,4          ; 指向键盘中断向量
        MOV     AX,KEYIP      ; 恢复原IP
        STOSW
        MOV     AX,KEYCS      ; 恢复原CS
        STOSW
        STI
        RET
KBPRG   ENDP                ; 返回DOS
KBINT   PROC                ; 键盘中断服务函数
        PUSH    AX
        PUSH    BX
        PUSH    SI
```


| | | | | | | |
|-------------|----------------|----------------|-------------|------------------|----------------|--------------|
| PUSH | DS | | ADD | BX,2 | | |
| STI | | | CMP | BX,OFFSET | BUFF+32 | |
| MOV | AX,DATA | | JNE | KBINT1 | | |
| MOV | DS,AX | | LEA | BX,BUFF | | |
| IN | AL,60H | KBINT1: | CMP | BX,BTOP | | |
| PUSH | AX | | JE | KBINT2 | | |
| IN | AL,61H | | MOV | BTLL,BX | | |
| OR | AL,80H | | MOV | AH,AL | | |
| OUT | 61H,AL | | LEA | BX,SCTAB | | |
| AND | AL,7FH | | XLAT | | | |
| OUT | 61H,AL | | MOV | [SI],AX | | |
| POP | AX | KBINT2: | MOV | AL,20H | POP | AX |
| TEST | AL,80H | | OUT | 20H,AL | IRET | |
| JNZ | KBINT2 | | POP | DS | KBINT | ENDP |
| MOV | BX,BTLL | | POP | SI | CODE | ENDS |
| MOV | SI,BX | | POP | BX | END | KBPRG |