



汇编语言程序设计

第五章 分支与循环程序设计

张华平 副教授 博士

Email: kevinzhang@bit.edu.cn

Website: <http://www.nlpir.org/>

@ICTCLAS张华平博士



大数据搜索与挖掘实验室 (wSMS@BIT)





- 采用结构化程序设计方法
- 3种基本结构
 - ⑩ 顺序结构
 - ⑩ 分支结构
 - ⑩ 循环结构





- 5.1 顺序结构程序设计
- 5.2 分支程序设计
- 5.3 循环程序设计



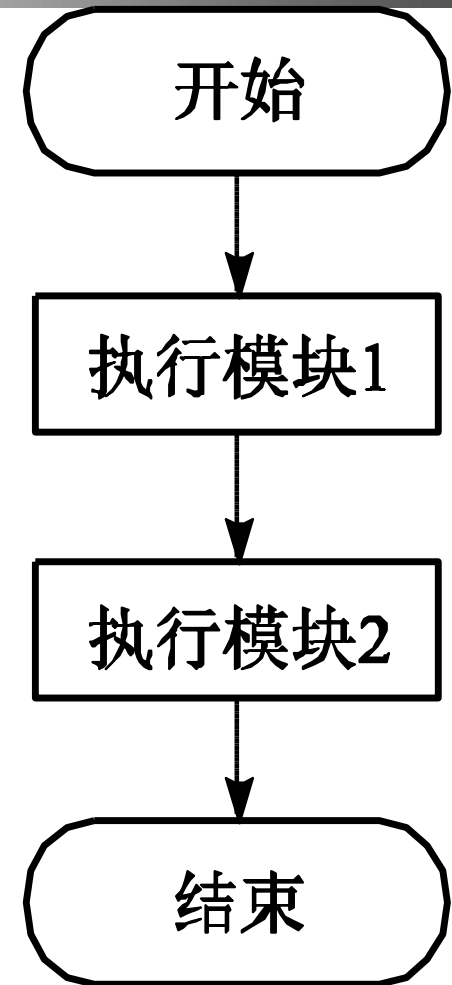


5.1 顺序结构程序设计

➤ 实现 $20 \times 11 \rightarrow M_WORD$:

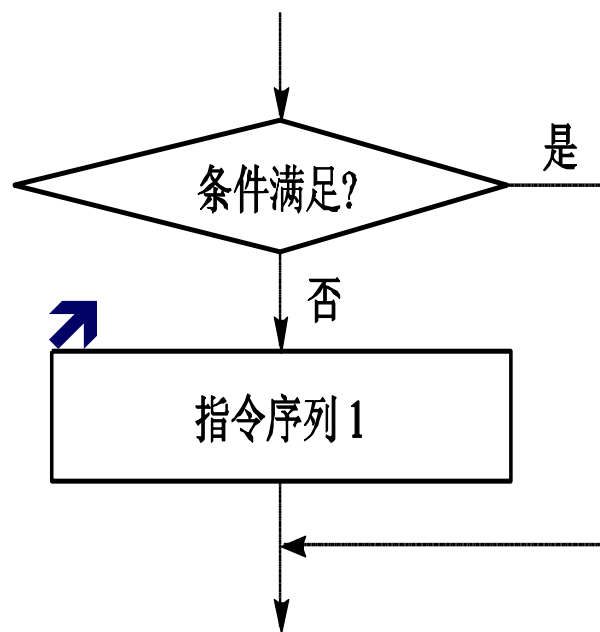


```
M_WORD    DW    ?  
  
MOV        AL,20  
MOV        BL,11  
MUL        BL  
MOV        M_WORD,AX
```

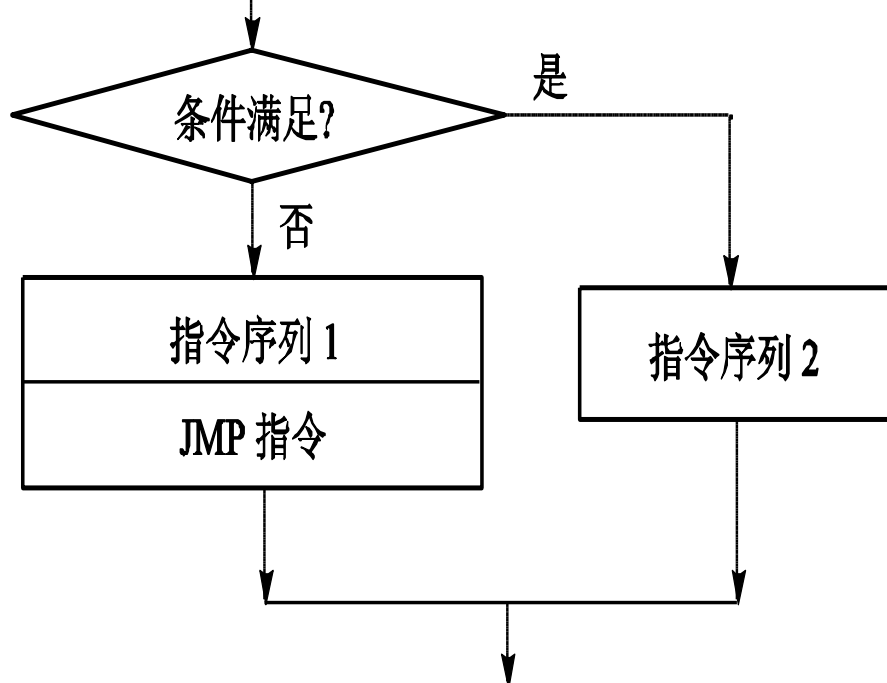


5.2 分支程序设计

➔ 需要根据不同条件进行不同的处理，此时需要设计分支程序。



(a) IF 结构; (b) IF_THEN_ELSE 分支结构



(a) IF 结构; (b) IF_THEN_ELSE 分支结构

➤ 1. IF 结构程序举例（单分支结构）

➤ 例5.1 求带符号数A和B的较大值 $MAXAB = MAX(A, B)$ 。 ; PROG0501

; PROG0501

MOV EAX, A

CMP EAX, B

JGE AlsLarger; 如果 $A \geq B$,

跳转到AlsLarger标号处

MOV EAX, B

AlsLarger:

MOV MAXAB, EAX



➤ 2. IF_THEN_ELSE 结构程序举例

**例5.3 求带符号数X的符号，如果
 $X \geq 0$ ，SIGNX置1，否则置为-1。**

;PROG0503





;PROG0503

```
X          SDWORD  -45
SIGNX      SDWORD      ?
MOV        SIGNX, 0
CMP        X, 0
JGE        XisPostive      ; X ≥ 0, 跳转
MOV        SIGNX, -1
JMP        HERE           ; 跳过 “MOV
```

SIGNX, 1” 语句

XisPostive:

```
          ;CMP X, 0
JE      XisZero
MOV          SIGNX, 1
JMP HERE
```

XisZero: MOV SIGNX, 0

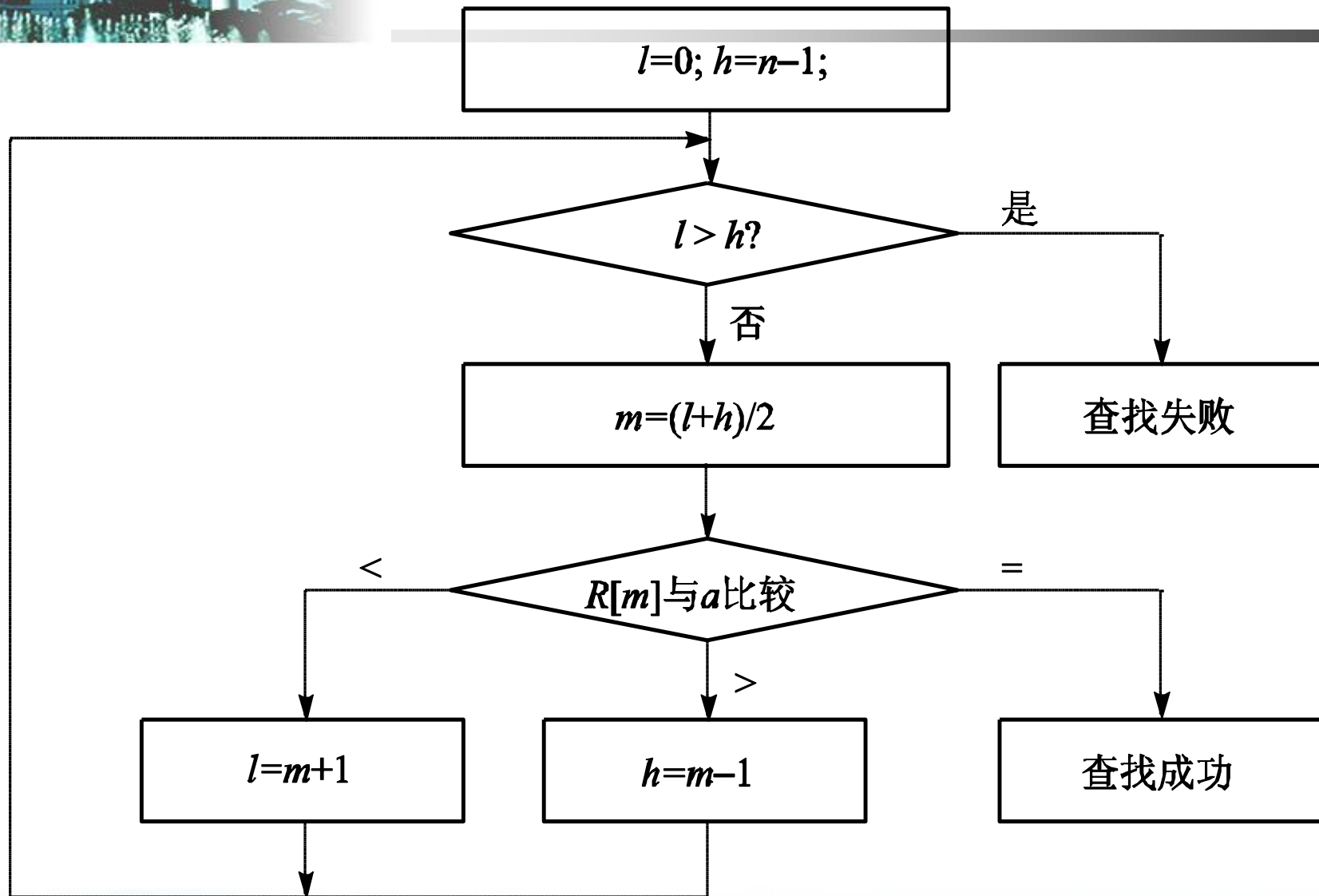
HERE:



例5.5 在升序数组中查找指定数。

- 算法分析：在有序数组中查找元素，使用折半查找的效率最高，平均比较次数为 $\log_2 n$ （顺序查找的次数 $n/2$ ）。本例使用折半查找法。设：
- 字节型数组为R
 - 元素个数为n
 - 要查找的数为a
 - 查找范围，以下界l和上界h表示
 - ✧ l和h是数组下标
 - ✧ 初始时下界为0上界为n-1，即整个数组
 - m是下界l和上界h的中点
 - 查找结束条件： $l > h$ ，意味无指定数





折半查找算法的流程





- 执行过程如下：
- (1) 先设定一个查找范围，以下界 l 和上界 h 表示。 l 和 h 是数组下标。初始时，下界为0，上界为 $n-1$ ，即查找范围是整个数组。
- (2) 如果下界 l 大于上界 h ，则查找范围为空，查找结束。在这种情况下，数组中没有 a ，算法结束。
- (3) 取下界 l 和上界 h 的中点 m , $m=(l+h)/2$ 。
- (4) 从数组的中点 m 处取出一个数 $R[m]$ ，和 a 进行比较。
- (5) 如果 $R[m]$ 等于 a ，则在数组中找到 a ，下标为 m 。算法结束。
- (6) 如果 $R[m]$ 大于 a ，中点上的数比 a 大，从中点到上界中的所有数都比 a 大，修改上界 h 为 $m-1$ 。然后跳转到第(2)步。
- (7) 如果 $R[m]$ 小于 a ，中点上的数比 a 小，从下界到中点中的所有数都比 a 小，修改下界 l 为 $m+1$ 。然后跳转到第(2)步。
- 每经过一次比较，查找范围就缩小一半。缩小查找范围的过程如图5-4所示。

;PROG0505



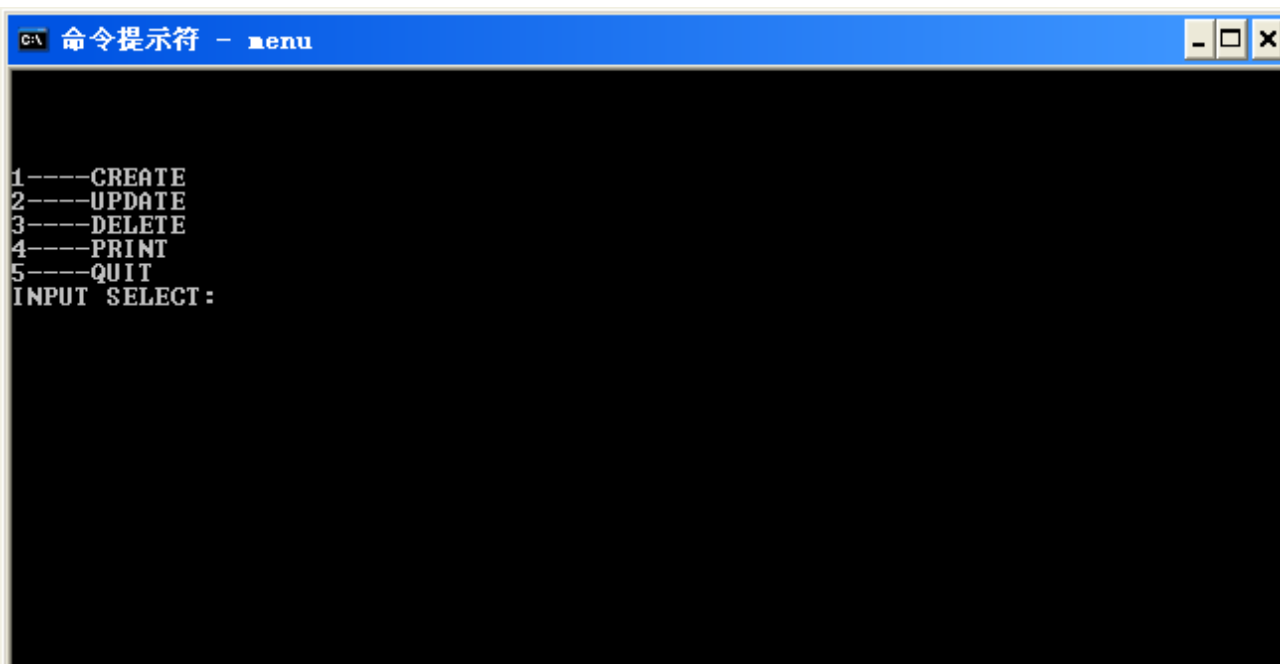


5.2.2 SWITCH_CASE结构分支程序设计

- 分支较少的IF_THEN_ELSE结构使用条件转移指令实现
- 分支较多时使用条件转移指令实现分支程序变得较复杂，不易读懂、扩充。多分支的SWITCH_CASE结构可用基于跳转表的方法实现
 - ✎ 实现的关键是先要构成跳转表
 - ✎ 然后用无条件间接转移指令实现跳转
- SWITCH_CASE结构可用于主控程序设计等

➤ 例. 编制一个管理文件的菜单程序，要求能够实现建立文件、修改文件、删除文件、显示文件和退出应用程序5个主控功能。首先在屏幕上显示5种功能，然后从键盘上输入数字1~5即可转入相应的功能，而输入其他字符则提示输入非法。若选择退出功能，则能正确返回；若选择其他功能，应能返回到主菜单。

➤ 屏幕显示为：



```
C:\ 命令提示符 - menu

1----CREATE
2----UPDATE
3----DELETE
4----PRINT
5----QUIT
INPUT SELECT:
```




算法分析：要能够转移到不同分支，必须提供各个分支的入口地址。对于SWITCH_CASE结构，由于分支众多，可以把各分支入口地址集中在一起构成一个地址表，把这个地址表称为跳转表。设建立文件分支入口标号为CR，修改文件分支入口标号为UP，删除文件分支入口标号为DE，显示文件分支入口标号为PR，退出分支入口标号为QU，则该跳转表如下所示：



```
JMPTAB                                DD    OFFSET CR
;跳转表
```



```
DD    OFFSET UP
```



```
DD    OFFSET DE
```

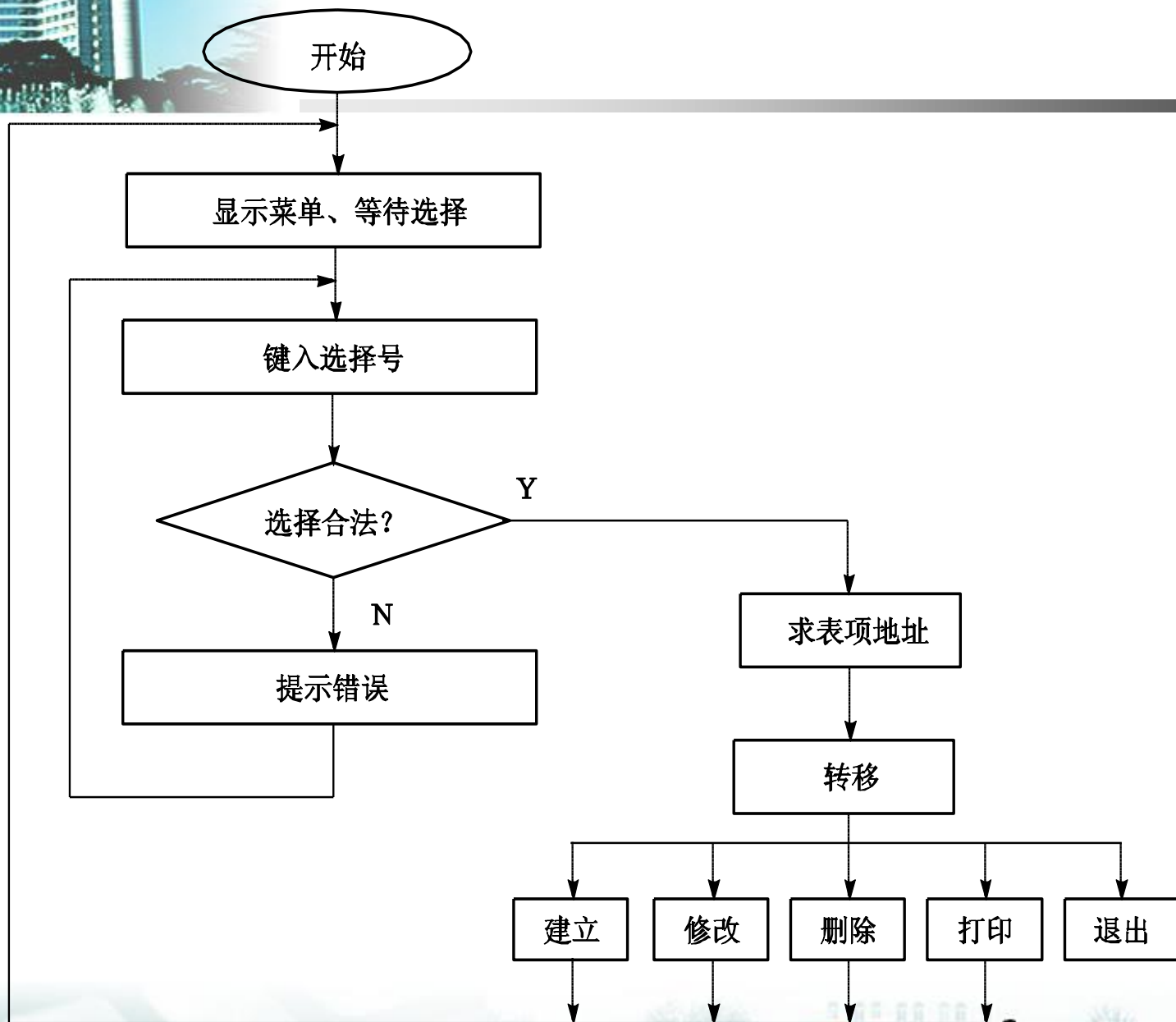


```
DD    OFFSET PR
```



```
DD    OFFSET QU
```





菜单程序流程图



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY



说明：位移量是跳转表中所选项与表基址的距离。把所有功能号连续排列，设选择了K号功能，则：

⑩索引号 = K - 一起始功能号（例如功能号为1, 2, 3...N, 则索引号 = K - 1）

⑩位移量 = 索引号 × 每项入口地址占用的字节数。

⑩表项地址 = 表基址 + 位移量。一旦得到了表项地址，使用无条件间接转移指令实现转移即可





◎用无条件间接转移指令实现SWITCH_CASE转

- 在32位程序中，地址表用DD定义，用段内间接转移指令实现跳转
- 在16位程序中，若跳转表是用DW定义的，则用段内间接转移指令。地址表用DD定义，则用段间间接转移指令

◎本例使用以下指令得到表项地址，并实现转移

```
mov    eax, JmpTab[ebx*4]
```

```
jmp    eax
```

也可以使用 `jmp JmpTab[ebx*4]` 指令等实现





5.3 循环程序设计

➤ 循环两种基本结构

● DO_WHILE结构

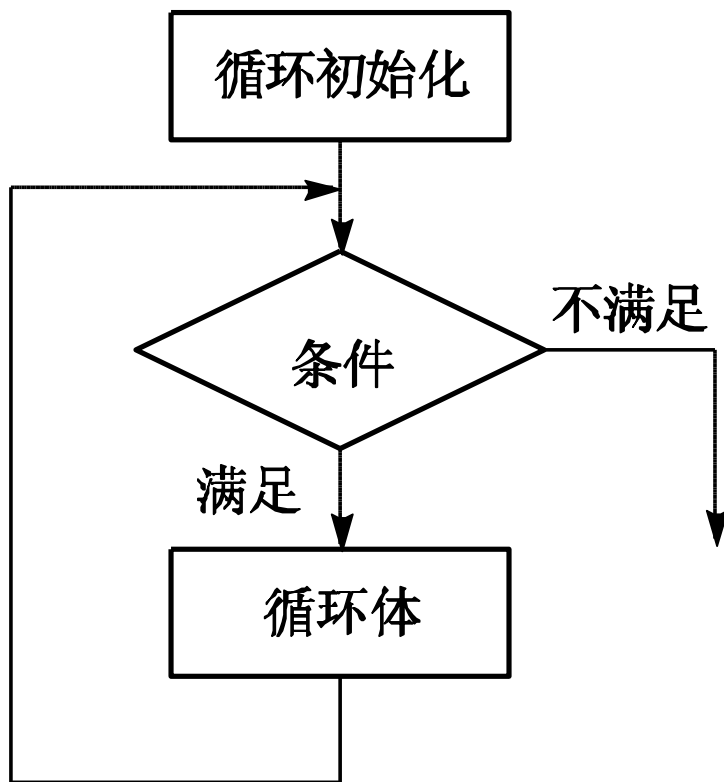
✎ 先判断后执行结构，把对循环控制条件的判断放在循环的入口，先判断控制条件，若满足控制条件就执行循环体，否则退出循环。

● DO_UNTIL结构

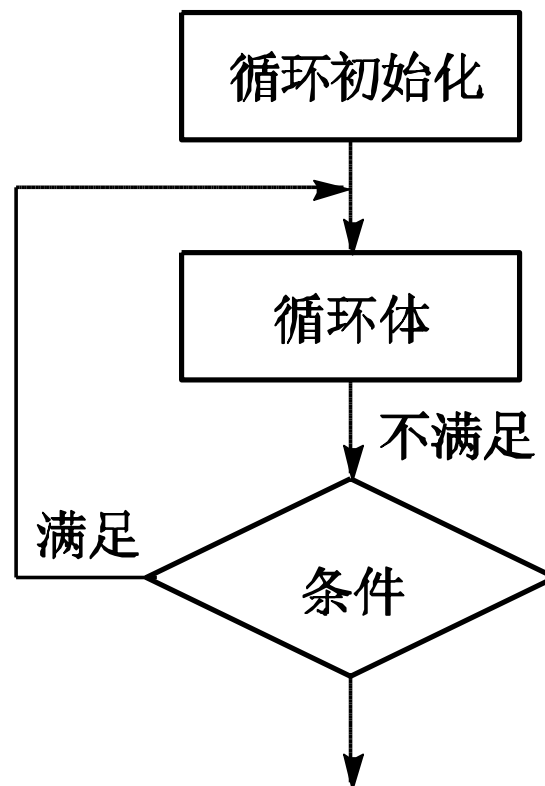
✎ 先执行后判断结构，先执行循环体然后再判断控制条件，若满足控制条件则继续执行循环体，否则退出循环

- 两种结构一般可以随习惯使用，但在初始循环次数可能为0的情况下则必须使用DO_WHILE结构





(a)



(b)

基本循环结构





- 无论使用哪种循环结构，循环程序一般应包括以下几个部分。
- (1) **循环初始化**。它包括设置循环次数的初始值、地址指针的初始设置等。
- (2) **循环体**。这是循环工作的主体，包括要重复执行的操作，以及循环的修改部分。修改部分包括地址指针的修改、循环控制条件的修改等。
- (3) **循环控制部分**。它是控制循环的关键，判断循环条件满足与否。例如判断循环次数是否为0等。





5.3.1 单重循环程序设计

➤ 例5.10 计算 n 的阶乘。

➤ 算法分析：阶乘（factorial）计算的公式为： $n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$ 。因此，需要循环 n 次，每次循环中完成一次乘法。

;PROG0510





;PR000510

. 386

.model flat, stdcall

option casemap:none

includelib

msvcrt.lib

printf

PROTO C :dword, :vararg

.data

Fact dword ?

N equ 6

szFmt byte 'factorial (%d)=%d', 0ah, 0;输出结果格式字符串

.code

start:

mov ecx, N;循环初值

mov eax, 1;Fact初值

e10:

imul eax, ecx;Fact=Fact*ECX

loop e10;循环N次

mov Fact, eax;保存结果

invoke printf, offset szFmt, N, Fact;打印结果

ret

end start





5.3.2 多重循环程序设计

- 有些复杂问题使用单重循环可能无法解决，此时就需要设计多重循环程序。
- 在多重循环的程序中，**内层循环嵌套于外层循环**，循环的嵌套层次没有限制。
- **各层循环都有各自的循环次数、循环体、循环结束条件，相互之间不能干扰、交叉。**





➤ DELAY

PROC

➤ MOV

BL, 20 ;置外循环次数

DELAY10:

MOV CX, 2801 ;置内循环次数

WT:

LOOP WT ;内循环体

DEC BL ;修改外循环次数

JNZ DELAY10 ;外循环控制

RET

➤ DELAY

ENDP





➤ 冒泡排序



感谢关注聆听！



张华平

Email: kevinzhang@bit.edu.cn

微博: @ICTCLAS张华平博士

实验室官网:

<http://www.nlpir.org>



大数据千人会

