

循环语句的机器级表示

用条件转移指令实现循环

```
int result = 0;
for(int i=1;i<=100;i++) {
    result +=i;
} //求 1+2+3+...+100
```

```
int i=1;
int result = 0;
while(i<=100) {
    result +=i;
    i++;
} //求 1+2+3+...+100
```

```
mov eax,0    #用 eax 保存 result, 初值为0
mov edx,1    #用 edx 保存 i, 初始值为1
cmp edx,100  #比较 i和100
jg L2        #若 i>100, 转跳到 L2 执行
L1:          #循环主体
add eax,edx  #实现 result +=i
inc edx      #inc 自增指令, 实现 i++
cmp edx,100  #i和100
jle L1       #若 i<=100, 转跳到 L1 执行
L2:          #跳出循环主体
```

C语言

用条件转移指令实现循环

```
int result = 0;
for(int i=1;i<=100;i++) {
    result +=i;
} //求 1+2+3+...+100
```

用条件转移指令实现循环，需要4个部分构成：

①循环前的初始化

②是否直接跳过循环？

③循环主体

④是否继续循环？

③循环主体

④是否继续循环？

①循环前的初始化

②是否直接跳过循环？

```
mov eax,0    #用 eax 保存 result, 初值为0
mov edx,1    #用 edx 保存 i, 初始值为1
cmp edx,100  #比较 i和100
jg L2        #若 i>100, 转跳到 L2 执行
L1:          #循环主体
add eax,edx  #实现 result +=i
inc edx      #inc 自增指令, 实现 i++
cmp edx,100  #i和100
jle L1       #若 i<=100, 转跳到 L1 执行
L2:          #跳出循环主体
```

用loop指令实现循环

```
for(int i=500;i>0;i--) {  
    做某些处理;  
} //循环500轮
```

理论上，能用 `loop` 指令实现的功能一定能用条件转移指令实现

使用 `loop` 指令可能会使代码更清晰简洁

补充： `loopx` 指令——如 `loopenz`, `loopz`
`loopenz`——当 `ecx!=0` && `ZF==0` 时，继续循环
`loopz`——当 `ecx!=0` && `ZF==1` 时，继续循环

```
mov ecx,500    #用ecx作为循环计数器  
Looptop:       #循环的开始  
...  
做某些处理  
...  
loop Looptop   #ecx--, 若ecx!=0, 跳转到Looptop
```

等价于：
`dec ecx`
`cmp ecx,0`
`jne Looptop`