

P138-142

a. 跳转指令

<code>movq \$0,%rax</code>	<i>Set %rax to 0</i>
<code>jmp .L1</code>	<i>Goto .L1</i>
<code>movq (%rax),%rdx</code>	<i>Null pointer dereference (skipped)</i>
<code>.L1:</code>	
<code>popq %rdx</code>	<i>Jump target</i>

指令 `jmp .L1` 会导致程序跳过 `movq` 指令。可以直接跳转，目标是编码，比如上图中的“`jmp .L1`”；也可以是间接跳转，目标是从寄存器或内存位置中读出的，比如“`jmp *%rax`”。

当然也有带条件的跳转指令，根据条件码的组合，或跳转，或执行下一条指令。

```
1    movq    %rdi, %rax
2    jmp     .L2
3    .L3:
4    sarq    %rax
5    .L2:
6    testq   %rax, %rax
7    jg      .L3
8    rep; ret
```

汇编器产生的“`.o`”格式的反汇编版本如下：

1	0:	48 89 f8	mov	%rdi,%rax	
2	3:	eb 03	jmp	8 <loop+0x8>	指明跳转目标
3	5:	48 d1 f8	sar	%rax	
4	8:	48 85 c0	test	%rax,%rax	
5	b:	7f f8	jg	5 <loop+0x5>	
6	d:	f3 c3	repz retq		

b. 用条件控制来实现分支

- 将条件表达式翻译成机器码，最常用的方式就是结合条件与跳转。
- 汇编器会插入条件和无条件分支，保证能够执行正确的代码块。