北京大学
PEKING UNIVERSITY

Peking University

Machine-Level Programming II: Control

汇报人 张立恒

2023年9月27日

目录 CONTENTS

北京大学
PEKING UNIVERSITY

01
课程内容总结

课程内容总结

课程内容总结

主要内容：

- 条件码（各种算术、逻辑运算指令、CMP、SET如何设置条件码）
- SET指令（对应的条件码组合）
- JMP指令（寻址方式）
- 条件语句（条件分支、条件传送）
- 循环语句（do-while,while,for）
- switch语句（跳转表的结构）

北京大学
PEKING UNIVERSITY

02

课程重点内容回顾

条件码

课程重点内容回顾

北京大学 PEKING UNIVERSITY

条件码寄存器	ADD指令下如何设置条件码
CF	(unsigned)t < (unsigned)a
ZF	t == 0
SF	t < 0
OF	(a<0 == b<0) && (t<0 != a<0)

可以换成(t>0 != a>0)吗?

或(a>0 && b>0 && t<0) || (a<0 && b<0 && t>=0)

overflow结果范围是(-pow(2,n-1), 0]吗?

SUB指令下(a>0 && b<0 && t<0) || (a<0 && b>0 && t>0)

注意： 移位操作把CF设置为最后一个被移出的位，OF设置为0

INC和DEC指令不改变CF

例题

课程重点内容回顾

北京大学 PEKING UNIVERSITY

（2016期中1）在下列指令中，其执行会影响条件码中的 CF 位的是：

A. jmp NEXT

B. jc NEXT

C. inc %bx

D. shl $1,%ax

设置条件码 CMP & TEST

课程重点内容回顾

只设置条件码，不改变其它寄存器

CMP　　SUB

TEST　AND

访问条件码

课程重点内容回顾

北京大学 PEKING UNIVERSITY

SET指令

setl　D　　D <- SF ^ OF

例题

课程重点内容回顾

2、条件码描述了最近一次算术或逻辑操作的属性。下列关于条件码的叙述中，哪一个是不正确的？

A. set 指令可以根据条件码的组合将一个字节设置为 0 或 1

B. cmp 指令和 test 指令可以设置条件码但不更改目的寄存器

C. leaq 指令可以设置条件码 CF 和 OF

D. 除无条件跳转指令 jmp 外，其他跳转指令都是根据条件码的某种组合跳转到标号指示的位置

跳转指令

课程重点内容回顾

北京大学 PEKING UNIVERSITY

一、无条件跳转

1. 直接跳转 jmp Label

2. 间接跳转 jmp *Operand

二、有条件跳转

条件与SET指令相同

三、机器代码对跳转目标的编码方式

1. 绝对地址

2. PC相对寻址　　地址偏移量用1、2或4字节补码表示

例题

课程重点内容回顾

（2019期中6）X86-64 指令提供了一组条件码寄存器；其中 ZF 为零标志，ZF=1 表示最近的操作得出的结构为 0；SF 为符号标志，SF=1 表示最近的操作得出的结果为负数；OF 为溢出标志，OF=1 表示最近的操作导致一个补码溢出（正溢出或负溢出）。当我们在一条 cmpq 指令后使用条件跳转指令 jg 时，那么发生跳转等价于以下哪一个表达式的结果为 1？

A. ~(SF ^ OF) & ~ZF

B. ~(SF ^ OF)

C. SF ^ OF

D. (SF ^ OF) | ZF

例题

课程重点内容回顾

北京大学 PEKING UNIVERSITY

（2016期中3）在如下代码段的跳转指令中，目的地址是：

```
400020: 74 F0                   je _________
400022: 5d                      pop %rbp
```

A. 400010 B. 400012 C. 400110 D. 400112

条件分支（条件跳转）&条件传送（条件赋值）

课程重点内容回顾


```
  ntest = !Test;
  if (ntest) goto Else;
  val = Then_Expr;
  goto Done;
Else:
  val = Else_Expr;
Done:
  . . .
```

```
result = Then_Expr;
eval = Else_Expr;
nt = !Test;
if (nt) result = eval;
return result;
```

注意：条件传送指令CMOV

1. 源值从内存或寄存器中读取
2. 不支持单字节的条件传送

例题

课程重点内容回顾

北京大学 PEKING UNIVERSITY

（2017期中5）在下列关于条件传送的说法中，正确的是：

A. 条件传送可以用来传送字节、字、双字、和 4 字的数据

B. C 语言中的“?:”条件表达式都可以编译成条件传送

C. 使用条件传送总可以提高代码的执行效率

D. 条件传送指令不需要用后缀（例如 b, w, l, q）来表明操作数的长度

注意：不适合/不能编译成条件传送的情况

循环

课程重点内容回顾

北京大学 PEKING UNIVERSITY

do-while

```
loop:
  Body
  if (Test)
    goto loop
```

while
jump to middle

```
  goto test;
loop:
  Body
test:
  if (Test)
    goto loop;
done:
```

guarded do

```
  if (!Test)
    goto done;
loop:
  Body
  if (Test)
    goto loop;
done:
```

for

例题

课程重点内容回顾

（2021期中7）阅读下列 C 代码和在 x86-64 机器上得到的汇编代码：

```
int a[__A__][__B__];
for (int i = 0; i < __C__; i++)
a[i][__C__ - i] = 1;
```

```
leaq 40(%rdi), %rax
addq $440, %rdi
.L2:
movl $1, (%rax)
addq $40, %rax
cmpq %rdi, %rax
jne .L2
```

假设 a 的地址初始时放在%rdi 中，假设程序正常运行且没有发生越界问题，则 C 代码中的 A、B、C 处应分别填：

A. 10、11、10

B. 9、11、9

C. 11、10、10

D. 11、11、11

switch语句

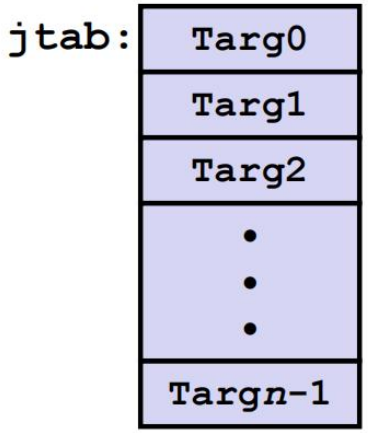
课程重点内容回顾

北京大学 PEKING UNIVERSITY

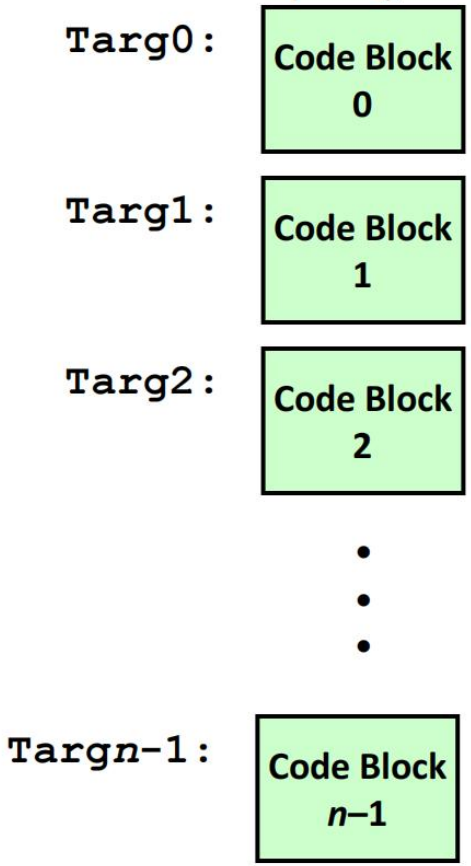
Switch Form

```
switch(x) {
  case val_0:
    Block 0
  case val_1:
    Block 1
    • • •
  case val_n-1:
    Block n–1
}
```

Jump Table



Jump Targets



Translation (Extended C)

```
goto *JTab[x];
```

适合用跳转表的情况

例题

课程重点内容回顾

（2014期中8）对简单的 switch 语句常采用跳转表的方式实现，在 x86-64 系统中，下述最有可能正确的 switch 分支跳转汇编指令为哪个？

A. jmp .L3(,%eax,4)

B. jmp .L3(,%eax,8)

C. jmp *.L3(,%eax,4)

D. jmp *.L3(,%eax,8)

例题

课程重点内容回顾

（2015年期中8）假设某条 C 语言 switch 语句编译后产生了如下的汇编代码及跳转表：

```
movl 8(%ebp), %eax
subl $48, %eax
cmpl $8, %eax
ja .L2
jmp *.L7(, %eax, 4)
.L7:
.long .L3      .long .L2      .long .L2      .long .L5.      long .L4
.long .L5      .long .L6      .long .L2      .long .L3
```

在源程序中，下面的哪些（个）标号出现过：

A. ‘2’, ‘7’　　B. 1　　C. ‘3’　　D. 5

其它例题

课程重点内容回顾

（2014期中五）阅读下面的汇编代码，根据汇编代码填写 C 代码中缺失的部分，然后描述该程序的功能。

```
pushl %ebp
movl %esp,%ebp
movl $0x0, %ecx
cmpl $0x0, 8(%ebp)
jle .L1
.L2
movl $0x0, %edx
movl 8(%ebp), %eax
divl $0x0a
addl %edx, %ecx
movl %eax, 8(%ebp)
cmpl $0x0, 8(%ebp)
jg .L2
.L1
movl 0x0, %edx
movl %ecx, %eax
divl 0x3
cmpl 0x0, %edx
jne .L3
movl 0x1, %eax
jmp .L4
.L3
movl 0x0, %eax
.L4
```

```
int fun(________x) {
int bit_sum = 0;
while (________) {
 ________;
 ________;
}
if (________)
return 1;
else
return 0;
}
```

谢谢观看

北京大学
PEKING UNIVERSITY

Machine-Level Programming II: Control

汇报人 张立恒

2023年9月27日