

Feb 24, 2016

Lecture 15

Jump instructions

①

A jump instruction can cause the execution to switch to a completely new position in the program.
jump destination - Label (in assembly code).

jmp - unconditional jump.

1. Direct jump. - jump target encoded as part of the instruction.
eg. `jmp Label`
2. Indirect jump - jump target is read from a register or a memory location.
eg. `jmp *Operand`

eg. `jmp *%eax`

`jmp *(%eax)`

Conditional jumps

Jumps only when a condition is satisfied.
- has various types |||^r to "set" instruction.

Conditional jumps can only be direct. ⁽²⁾

eg. je label

jl label

Different conditional jumps

je, jne, js, jns,

jq, jge, jl, jle,


ja, jae, jb, jbe

Encoding jump targets

1. PC relative 

2. "absolute" address

SHOW CODE for jump instructions.

 Especially PC relative jumps!

(3)

Loops

do-while loop - most compilers generate loop code based on this form.

Do-While Loops

```
do  
    body-of-loop  
while (test-expr);
```

⇒

```
loop:  
    body-of-loop  
    t = test-expr;  
    if (t)  
        goto loop;
```

Factorial program - do-while version

C code

```
int fact_do(int n)  
{  
    int result = 1;  
    do {  
        result *= n;  
        n = n - 1;  
    } while (n > 1);  
    return result;  
}
```

Corresponding Assembly ^④ language code

Argument: n at $\%ebp + 8$

Registers: n in $\%edx$, result in $\%eax$.

`movl 8(%ebp), %edx`

`movl $1, %eax`

Get n .

set result = 1.

• L2:

`imull %edx, %eax`

`subl $1, %edx`

`cmpl $1, %edx`

`jg .L2`

`return result.`

loop:

`result *= n;`

`n--;`

`compare n:1`

`if >, goto loop.`

While Loops

```

while (test-expr)
  body-statement

```



```

if (!test-expr)
  goto done;
do
  body-stmt
  while (test-expr);
done:

```

Goto version:

```

t = test-expr;
if (!t)
  goto done;

```

```

loop:
  body-stmt
  t = test-expr;
  if (t)
    goto loop;

```

done:

For loops

```

for (init-expr; test-expr; update-expr)
  body-statement

```

(6)

For → while

init-expr;

while (test-expr) {

body-stmt

update expr;

}

For → do-while form :

init-expr;

if (! test-expr)
goto done;

do {

body-stmt .

update-expr;

} while (test-expr);

done :

For \rightarrow goto form: ⁽⁷⁾

init-expr;

t = test-expr;

if (!t)
goto done;

loop:

body-statement

update-expr;

t = test-expr;

if (t)
goto loop;

done:

Switch statements