

计算机系统设计平时作业四

朱浩泽 1911530 计算机科学与技术

If-else语句举例

```
int get_cont( int *p1, int *p2 )
{
    if ( p1 > p2 )
        return *p2;
    else
        return *p1;
}
```

p1和p2对应实参的存储地址分别为R[ebp]+8、R[ebp]+12，EBP指向当前栈帧底部，结果存放在EAX。

为何这里是“jbe”指令？

```
movl 8(%ebp), %eax //R[edx] ← M[R[ebp]+8], 即 R[edx]=p1
movl 12(%ebp), %edx //R[edx] ← M[R[ebp]+12], 即 R[edx]=p2
cmpl %edx, %eax    //比较 p1 和 p2, 即根据 p1-p2 结果置标志
jbe .L1            //若 p1 ≤ p2, 则转 L1 处执行
movl (%edx), %eax
jmp .L2            //无条件跳转到 L2 执行
.L1:
movl (%eax), %eax
.L2
```

1

当执行到 jbe 命令时，如果此时的 CF 标志位或者 ZF 标志位为1则跳转，其他不跳转，相当于小于等于命令。即执行到 cmp 命令时，如果此时的 edx 小于等于 eax，jbe 执行跳转指令。

我们使用如下程序测试单独的if语句

```
int main()
{
    int a, b;
    if(a > b) {
        a = 1;
        b = 2;
    }
}
```

使用gcc编译生成汇编代码如下：

```
movl    -8(%rbp), %eax
cmpl    -4(%rbp), %eax
jle     .L2
movl    $1, -8(%rbp)
movl    $2, -4(%rbp)
.L2:
```

如果使用大于命令则会生成如下汇编代码

```
    movl    -8(%rbp), %eax
    cmpl    -4(%rbp), %eax
    ja     .L1
    jmp     .L2
.L1:
    movl    $1, -8(%rbp)
    movl    $2, -4(%rbp)
.L2:
```

通过以上实验可以看出，使用jbe的相当于进行了一个小小的预判，其代码量明显小于使用ja指令的代码量，即如果if条件没有发生则直接跳过其指令执行else或下面的指令；如果发生直接执行结果然后进行跳转便可，处理器的流水线不会产生气泡造成资源浪费。