Homework 2

2018011365 张鹤潇

在gcc中，`__builtin_expect(X, Y)`表示期望$X = Y$. 具体而言，`__builtin_expect(!!(x),1)`表示期望`x`为真，而`__builtin_expect(!!(x),0)`表示期望`x`为假。`__builtin_expect`的主要作用是**帮助编译器判断条件跳转的预期值，避免因执行跳转指令造成时间浪费。**在处理器流水线中同时运行着多条指令，跳转会使已读入的指令作废，降低执行效率，因此要尽可能优化跳转指令。

具体分析如下，在使用`unlikely(a==2)`时，执行`else`分支的概率更大，关键部分汇编代码：

```
movl        %eax, %esi
movl        $3, %ecx  # if a == 2, then ++a = 3
imull      %eax, %esi  # b = a^2
cmpl        $2, %eax
leal        1(%rsi), %edx  # %edx = b++
je            .L3
# else
leal       -1(%rax), %ecx # a--
leal       -1(%rsi), %edx # %edx = b--
# if a == 2:
.L3:
leal        (%rcx,%rdx), %eax  # ret a + b
ret
```

此时执行`if`分支会比执行`else`分支多跳转一次。

在使用`likely(a==2)`时，执行`if`分支的概率更大，关键部分汇编代码：

```
movl        %eax, %ecx
imull      %eax, %ecx  # b = a^2
cmpl        $2, %eax
jne          .L2  # jump if a != 2
# if a == 2:
leal        1(%rcx), %eax  # b++
movl        $3, %edx # a++
.L3:
addl        %edx, %eax # ret = a + b
ret
# else
.L2:
leal       -1(%rax), %edx  # a--
leal       -1(%rcx), %eax  # b--
jmp          .L3
```

此时执行`else`分支会比执行`if`分支多跳转两次。