

关键汇编代码分析 (work3_1.s)

```
main:
    # ... 初始化代码 ...
    movl    $10, -8(%rbp)    # x = 10
    movl    $0, -4(%rbp)    # res = 0
    # ... printf和scanf调用 ...

    # Switch跳转表处理
    movl    -8(%rbp), %eax    # 加载x值
    subl    $100, %eax        # x - 100
    cmpl    $10, %eax        # 检查范围(0-10)
    ja      .L2               # 如果>10跳default
    mov     %eax, %eax
    leaq    .L4(%rip), %rdx    # 加载跳转表基址
    movslq  (%rdx,%rax,4), %rax # 获取偏移量
    addq    %rdx, %rax        # 计算跳转地址
    jmp     *%rax             # 跳转

.L4:  # 跳转表(10个条目)
    .long   .L3-.L4          # case 100 (x=100)
    .long   .L2-.L4          # default (x=101)
    .long   .L5-.L4          # case 102 (x=102)
    .long   .L6-.L4          # case 103 (x=103)
    .long   .L7-.L4          # case 104 (x=104)
    .long   .L8-.L4          # case 105 (x=105)
    .long   .L9-.L4          # case 106 (x=106)
    .long   .L10-.L4         # case 107 (x=107)
    .long   .L2-.L4          # 未使用 (x=108本应跳这里)
    .long   .L10-.L4         # case 108 (实际跳转到107)
    .long   .L2-.L4          # default (x=109)

.L3:  # case 100
    movl    $1, -4(%rbp)
    jmp     .L11
.L5:  # case 102
    movl    $2, -4(%rbp)
    jmp     .L11
.L6:  # case 103
    movl    $3, -4(%rbp)    # 注意没有break!
.L7:  # case 104
    addl    $4, -4(%rbp)    # res += 4
.L8:  # case 105
    addl    $5, -4(%rbp)    # res += 5
    jmp     .L11
.L9:  # case 106
    movl    $6, -4(%rbp)
    jmp     .L11
.L10: # case 107/108
    movl    $7, -4(%rbp)
```

```
        jmp      .L11
.L2:    # default
        movl     $10, -4(%rbp)
.L11:
        # ... printf调用 ...
```

跳转表详细说明

跳转表结构（基于x-100的索引）：

索引	对应x值	跳转目标	说明
0	100	.L3	case 100
1	101	.L2	default
2	102	.L5	case 102
3	103	.L6	case 103 (会fall through)
4	104	.L7	case 104
5	105	.L8	case 105
6	106	.L9	case 106
7	107	.L10	case 107
8	108	.L2(未使用)	编译器优化为使用索引9
9	108	.L10	实际处理case 108的位置
10	109	.L2	default

C代码与汇编的映射关系

C代码块	汇编标签	关键行为
case 100: res=1;break	.L3	movl \$1, -4(%rbp)
case 102: res=2;break	.L5	movl \$2, -4(%rbp)
case 103:	.L6	movl \$3, -4(%rbp) → 继续执行下一条
case 104: res+=4	.L7	addl \$4, -4(%rbp)
case 105: res+=5	.L8	addl \$5, -4(%rbp)
case 106: res=6;break	.L9	movl \$6, -4(%rbp)
case 107/108: res=7	.L10	movl \$7, -4(%rbp)
default: res=10	.L2	movl \$10, -4(%rbp)