P252-255

- Y86-64 程序

  代码比对：

```
1   long sum(long *start, long count)
2   {
3       long sum = 0;
4       while (count) {
5           sum += *start;
6           start++;
7           count--;
8       }
9       return sum;
10  }
```

x86-64 code

```
long sum(long *start, long count)
start in %rdi, count in %rsi
1  sum:
2    movl    $0, %eax        sum = 0
3    jmp     .L2             Goto test
4  .L3:                    loop:
5    addq    (%rdi), %rax    Add *start to sum
6    addq    $8, %rdi        start++
7    subq    $1, %rsi        count--
8  .L2:                    test:
9    testq   %rsi, %rsi      Test sum
10   jne     .L3             If !=0, goto loop
11   rep; ret                Return
```

Y86-64 code

```
long sum(long *start, long count)
start in %rdi, count in %rsi
1  sum:
2    irmovq $8,%r8       Constant 8
3    irmovq $1,%r9       Constant 1
4    xorq %rax,%rax      sum = 0
5    andq %rsi,%rsi      Set CC
6    jmp     test        Goto test
7  loop:
8    mrmovq (%rdi),%r10  Get *start
9    addq %r10,%rax      Add to sum
10   addq %r8,%rdi       start++
11   subq %r9,%rsi       count--.  Set CC
12 test:
13   jne     loop        Stop when 0
14   ret                 Return
```

图 4-6　Y86-64 汇编程序与 x86-64 汇编程序比较。Sum 函数计算一个整数数组的和。Y86-64 代码与 x86-64 代码遵循了相同的通用模式

  - x86-64 代码是由 GCC 编译器产生的，y86-64 类似，不同点：
    - 常数加载到寄存器，因为其在算术指令中不能使用立即数。
    - 从内存读取数值并与寄存器相加，需要两条指令，而 x86-64 只需要 addq 一条指令

- Y86-64 指令详情
  - pushq 指令会把栈指针-8，并且将一个寄存器值写入内存，