

实验 1：实用汇编程序

PB22030892 刘铠瑜

任务 1：斐波那契数列

fib_1.asm

```
.text
.globl main

main:
    li x4, 0      # 初始化x4为0（斐波那契数列的第0项）
    li x5, 1      # 初始化x5为1（斐波那契数列的第1项）
    li x3, 1      # 初始化x3为1（用于存储当前计算的斐波那契数）

loop:
    addi x2, x2, -1 # 将x2的值减1，相当于循环计数器
    beq x2, x0, end # 如果x2等于0，跳转到end结束循环
    add x3, x4, x5   # 计算x3 = x4 + x5（当前斐波那契数）
    mv x4, x5        # 将x5的值移动到x4（更新前一个数）
    mv x5, x3        # 将x3的值移动到x5（更新当前数）
    j loop           # 跳转回loop继续循环

end:
    li a7, 10        # 设置系统调用号10（exit）
    ecall            # 执行系统调用，程序结束
```

任务 2：大整数处理

fib_2.asm

```

.text
.globl main

main:
    li x4, 1          # x4 = 1 (用于存储当前斐波那契数的低位部分)
    li x3, 0          # x3 = 0 (用于存储当前斐波那契数的高位部分)
    li x6, 0          # x6 = 0 (用于存储前一个斐波那契数的低位部分)
    li x5, 0          # x5 = 0 (用于存储前一个斐波那契数的高位部分)
    li x8, 1          # x8 = 1 (当前斐波那契数的低位部分)
    li x7, 0          # x7 = 0 (当前斐波那契数的高位部分)

loop:
    addi x2, x2, -1    # x2 = x2 - 1 (递减循环计数器)
    beq x2, x0, end    # 如果 x2 == 0, 则跳转到 end, 结束循环

    # 计算下一个斐波那契数的低位部分
    add x4, x8, x6      # x4 = x8 + x6 (当前低位 + 前一个低位)
    sltu x9, x4, x8      # x9 = (x4 < x8) ? 1 : 0 (检查是否有低位溢出)

    # 计算下一个斐波那契数的高位部分
    add x3, x7, x5      # x3 = x7 + x5 (当前高位 + 前一个高位)
    add x3, x3, x9      # x3 = x3 + x9 (加上低位溢出的进位)

    # 更新寄存器值, 为下一次循环做准备
    mv x6, x8          # x6 = x8 (将当前低位赋值给前一个低位)
    mv x5, x7          # x5 = x7 (将当前高位赋值给前一个高位)
    mv x8, x4          # x8 = x4 (将计算出的低位赋值给当前低位)
    mv x7, x3          # x7 = x3 (将计算出的高位赋值给当前高位)

    j loop             # 跳转到 loop, 继续下一次循环

end:
    li a7, 10          # a7 = 10 (设置退出系统调用的代码)
    ecall              # 调用系统退出程序

```

任务 3: 导出 COE 文件

fib_1.coe

```
memory_initialization_radix  = 16;
memory_initialization_vector =
00000213
00100293
00100193
fff10113
00010a63
005201b3
00500233
003002b3
fedff06f
00a00893
00000073
```

fib_2.coe

```
memory_initialization_radix  = 16;
memory_initialization_vector =
00100213
00000193
00000313
00000293
00100413
00000393
fff10113
02010463
00640233
008234b3
005381b3
009181b3
00800333
007002b3
00400433
003003b3
fd9ff06f
00a00893
00000073
```