**作业03 汇编指令序列分析**

彭程 2020011075

.LC1:

        .string "%d\n" //当成printf的参数用

main: （主函数）

        push    r12 //将r12的值压入栈

        pxor    xmm1, xmm1 //xmm1清零

        mov     r12d, 2 // r12d初始化为2，即局部变量i

.L9: （开平方计算和赋值）

        pxor    xmm0, xmm0 //xmm0清零

        cvtsi2sd        xmm0, r12d //将r12d转换成双精度浮点放入xmm0

        ucomisd xmm1, xmm0 //比较xmm1和xmm0大小

ja看的是~CF&~ZF，而uncomisd设置的是ZF，PF, and CF flags

        ja      .L18 //若xmm1比xmm0大，跳转到L18，即对负数开方

        sqrtsd  xmm0, xmm0 //对xmm0开方并将结果放回xmm0

.L4:（sqr赋值和特殊情况处理）

        cvttsd2si       esi, xmm0 //将xmm0中的双精度浮点转为有符号整型放入esi，esi放的即为sqr的值

        cmp     esi, 1 //比较esi和1的大小

        jle     .L5 //若esi小于等于1，跳转到L5

        test    r12b, 1 //将r12b和1与，即看r12b是不是0

        je      .L6 //结果为0，即r12b为0，跳转到L6

        mov     ecx, 2 //ecx初始化为2，即局部变量j

        jmp     .L7 //跳转到L7

.L8:（判断余数是否为0）

        mov     eax, r12d //将eax的值置为r12d的值

        cdq //将edx赋成eax的符号位，即rdx置0

The CDQ instruction copies the sign (bit 31) of the value in the EAX register into every bit position in the EDX register.

        idiv    ecx //将R[rdx]:R[rax]除以R[ecx]

        test    edx, edx //判断edx是否为0，即eax的符号

        je      .L6 //edx为0，即eax为正数，跳转L6

.L7:（内循环判断）

        add     ecx, 1 //ecx+1,即j++

        cmp     esi, ecx //比较esi和ecx，即比较sqr和j

        jge     .L8 //若esi≥ecx，即sqr≥j，跳转到L8

.L5:（不是合数，进入下一次外循环）

        add     r12d, 1 //r12d+1,即i++

        cmp     r12d, 101 //比较r12d和101

        jne     .L9 //若r12d≠101，跳转到L9

.L22:（程序结束）

        xor     eax, eax //eax清零

        pop     r12 //将栈顶的数据送入r12

        ret

.L6:（是合数并打印，进入下一次外循环）

        mov     esi, r12d //将r12d的值赋给esi作为参数1

        mov     edi, OFFSET FLAT:.LC1 //将LC1指令赋给edi作为参数2

        xor     eax, eax //清空eax

        add     r12d, 1 //r12d+1,即i++

        call    printf //调用printf函数

        pxor    xmm1, xmm1 //xmm1置0

（怀疑是printf函数改动了xmm1，故此处要将xmm1清零）

        cmp     r12d, 101 //比较r12d和101

        jne     .L9 //若r12d≠101，跳转到L9

        jmp     .L22 //跳转到L22

.L18:（被开方数是负数的特殊处理）

        call    sqrt //调用sqrt库函数进行错误处理

        pxor    xmm1, xmm1 //将xmm1清零

（怀疑是sqrt函数改动了xmm1，故此处要将xmm1清零）

        jmp     .L4 //跳回L4