

课内实验-RV32I冒泡排序

李毅PB22051031

题目：基于RV32汇编，设计冒泡排序程序，并用Ripes工具调试执行。

1.实现思路

通过两层循环，对数组中相邻元素逐一比较，每循环一次，将本轮循环中最大（最小）的元素放在高地址处。若有N个元素，则外层循环N-1次，外层第i次循环内层循环N-1-（i-1）次。

2.核心代码

```
Loop1: #外层循环
    bge x10,x15,Done
    add x9,x3,x0 # 数组地址
    add x11,x10,x0 # j=i,内层循环N-i次，这样赋值可以确定内层循环次数
Loop2: #内层循环
    bge x11,x15,Loop1a
    lw x12, 0(x9)
    lw x13, 4(x9)
    bge x12,x13,skip #大于则跳过
    sw x13,0(x9) #数据调换
    sw x12,4(x9)
skip:
    addi x9, x9,4 #addr++
    addi x11,x11,1 #j++
    beq x0,x0,Loop2
Loop1a:
    addi x10,x10,1 #i++
    beq x0,x0,Loop1
```

3.内存数据段

排序前：

Address	Word	Byte 0	Byte 1	Byte 2	Byte 3
0x10000024	0x00000001	0x01	0x00	0x00	0x00
0x10000020	0x00000003	0x03	0x00	0x00	0x00
0x1000001c	0x0000000c	0x0c	0x00	0x00	0x00
0x10000018	0x00000005	0x05	0x00	0x00	0x00
0x10000014	0x0000000a	0x0a	0x00	0x00	0x00
0x10000010	0x00000007	0x07	0x00	0x00	0x00
0x1000000c	0x00000004	0x04	0x00	0x00	0x00
0x10000008	0x00000009	0x09	0x00	0x00	0x00
0x10000004	0x00000002	0x02	0x00	0x00	0x00
0x10000000	0x00000002	0x02	0x00	0x00	0x00

排序后：

Address	Word	Byte 0	Byte 1	Byte 2	Byte 3
0x10000024	0x00000001	0x01	0x00	0x00	0x00
0x10000020	0x00000002	0x02	0x00	0x00	0x00
0x1000001c	0x00000002	0x02	0x00	0x00	0x00
0x10000018	0x00000003	0x03	0x00	0x00	0x00
0x10000014	0x00000004	0x04	0x00	0x00	0x00
0x10000010	0x00000005	0x05	0x00	0x00	0x00
0x1000000c	0x00000007	0x07	0x00	0x00	0x00
0x10000008	0x00000009	0x09	0x00	0x00	0x00
0x10000004	0x0000000a	0x0a	0x00	0x00	0x00
0x10000000	0x0000000c	0x0c	0x00	0x00	0x00

实现了降序排列（从低地址到高地址）

4.控制台输出

```
past:2 2 9 4 7 10 5 12 3 1
now:12 10 9 7 5 4 3 2 2 1
Program exited with code: 0
```

5.典型问题及解决

由于本实验中，我先自主实现了冒泡排序，再参考ripes提供的例程实现了控制台输出，在实现冒泡排序时直接调用寄存器，而在实现了控制台输出时参考例程使用了伪码调用寄存器。于是发生了寄存器冲突问题。后使用栈指针x2和函数调用解决。也可以通过全部使用伪码调用寄存器解决冲突。

6.完整代码

```
.data
    array:.word 2,2,9,4,7,10,5,12,3,1 # PB22051031李毅,10个数

    str1: .string " "
    str2: .string "\n"
    past: .string "past:"
    now: .string "now:"

.text
    lui x3, 0x10000
    addi x15, x0, 9 # N=10
    add x9,x3,x0 # 数组地址

    la a0, past
    li a7, 4
    ecall
    jal x1,print

    addi x10, x0, 0 # i
Loop1: #外层循环
    bge x10,x15,Done
    add x9,x3,x0 # 数组地址
    add x11,x10,x0 # j=i,内层循环N-i次，这样赋值可以确定内层循环次数
Loop2: #内层循环
    bge x11,x15,Loop1a
    lw x12, 0(x9)
    lw x13, 4(x9)
    bge x12,x13,skip #大于则跳过
    sw x13,0(x9) #数据调换
    sw x12,4(x9)
skip:
    addi x9, x9,4 #addr++
    addi x11,x11,1 #j++
    beq x0,x0,Loop2
Loop1a:
    addi x10,x10,1 #i++
    beq x0,x0,Loop1

Done:

    la a0, str2
    li a7, 4
    ecall

    la a0, now
    li a7, 4
    ecall
    jal x1,print
```

```
addi x17, x0, 10
ecall # x17=10表示程序退出
```

print:

```
addi sp, sp, -4
sw x10, 0(sp) #保存数据

la a0, array
addi t0, a0, 0 #t0存储array地址
addi t2, a0, 0 #copy of a0
li t1, 0 #t1=k=0
```

print_array:

```
lw a0, 0(t0) #取到寄存器
li a7, 1
ecall
la a0, str1
li a7, 4
ecall
addi t1, t1, 1
addi t0, t0, 4 #t0++
addi t2, x15, 1
blt t1, t2, print_array #(N为排序时外层循环次数，数组遍历时循环次数为N+1)
addi a0, t2, 0
ecall

lw x10, 0(sp)
addi sp, sp, 4

jalr x0, x1, 0
```