中山大学数据科学与计算机学院本科生实验报告

（2019 学年秋季学期）

课程名称：**计算机组成原理实验**　　任课教师：郭雪梅　　助教：丁文、汪庭葳

年级&班级	2019 级 04 班	专业(方向)	计算机科学与技术（超算方向）
学号	19335112	姓名	李钰
电话	19847352856	Email	1643589912@qq.com
开始日期		完成日期	

一、 实验题目

计算机结构与组成 MIPS 汇编程序

二、实验目的

熟悉 MARS 仿真器的使用，学习用它来运行和调试程序，并将所学理论知识合理应用。

三、实验内容及实验结果

<练习一>

a. .data, .word, .text 指示器（directives）的含义是什么(即, 在每段中放入什么内容)?

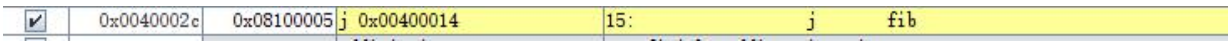
答：①.data: Subsequent items stored in data segment at next available address(放入在可用地址的下一个数据段中的后续项)

②.word:store the listed values as 32 bit words boundary(将列出的值存储为 32 位字边界）

③.text: subsequent items (instructions) stored in text segment at next available address(后续项目（指令）存储在下一个可用地址的文本段中)

b. 在 MARS 中如何设置断点 breakpoint? 请在第 15 行设置断点，并在所有问题解答完后，将此结果给老师检查。

答：点击上方菜单栏中 run 选中 Assemble ,在弹出的 Text Segment 框中第一列 Bkpt 中选择第 15 行前打勾即为设置断点。



c. 在程序运行到断点处停止时，如何继续执行? 如何单步调试代码?

答：继续执行：run the current program ; 单步调试代码：run one step at a time

d. 如何知道某个寄存器 register 的值是多少? 如何修改寄存器的值.

答： Exeute 页面中右侧 registers 栏中可查看寄存器中的值是多少；双击 value 下对应的要修改的位置即可直接修改值。

e. n 存储在内存中的哪个地址? 通过修改此内存处的值来计算第 13 个 fib 数.

答：n 储存在第 11 号寄存器中，第十三个 fib 数为 233。

f. 16 和 18 行使用了 syscall 指令. 其功能是什么，如何使用它? (提示: syscall 在 Help 中有说明!如何英文不是太好，可以一边运行，一边看效果，来体会其用途)

答： syscall 系统调用，调用$v0 中指定的系统调用

第 1 步 在寄存器$v0 中加载服务编号。

第 2 步 加载指定的$a0、$a1、$a2 或$f12 的参数值（如果有）。

第 3 步 发出 SYSCALL 指令。

第 4 步 从指定的结果寄存器中检索返回值（如果有）。

<练习二>

代码：

```
        .data

        .text

main:

   addi $s2, $zero, 2          # 将$s2 设为 flag，值存入 2

Loop:

   subi $s2, $s2, 1            # 每循环一次，flag($s2)就减一

   add $t0,$s0,$zero

   add $t1,$s1,$zero

   add $t2,$t0,$t1
```

```
    add $t3,$t1,$t2

    add $t4,$t2,$t3

    add $t5,$t3 , $t4

    add $t6,$t4 , $t5

    add $t7,$t5 , $t6

    bne $s2, $zero, Loop      # 把 flag（$s2)和 0 比较，如果不相等，继续循环，
如果相等，则结束，最终实现循环两次

  syscall
```

<练习三>

```
          .data
source:   .word   3, 1, 4, 1, 5, 9, 0
dest:     .word   0, 0, 0, 0, 0, 0, 0
countmsg: .asciiz " values copied. "
          .text
main:   la      $a0,source
        la      $a1,dest
loop:   lw      $v1, 0($a0)     # read next word from source
        addiu   $v0, $v0, 1     # increment count words copied
        sw      $v1, 0($a1)     # write to destination
        addiu   $a0, $a0, 4     # advance pointer to next source
        addiu   $a1, $a1, 4     # advance pointer to next dest
        bne     $v1, $zero, loop# loop if word copied not zero
loopend:
        move    $a0,$v0         # $a0 <- count
```

```
        jal     puti            # print it
        la      $a0,countmsg    # $a0 <- countmsg
        jal     puts            # print it
        li      $a0,0x0A        # $a0 <- '\n'
        jal     putc            # print it
finish:
        li      $v0, 10         # Exit the program
        syscall


puti:
        li $v0 1
        syscall
        jr $ra
puts:
        li $v0 4
        syscall
        jr $ra
putc:
        li $v0 11
        syscall
        jr $ra
```

四、实验感想

通过本次实验，我初步了解并掌握了 Mars 的应用，能成功利用 Mars 进行一些简单汇编语言程序的编写，学会了如何利用 Mars 来进行程序的调试，对错误的代码进行改正，学会了通过直接在寄存器里改值的方法来改变变量的值。