南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

计算机组成原理实验报告

姓名： 卫焱滨　　　　　　　　学号：11710823

一、实验目的

1.1 练习并熟练掌握基于 MIPS 的汇编语言的基本操作：包括加减乘除等算数计算、数据的装载（load）、系统命令（输出），变量定义和特定寄存器的使用（如 hi，lo）等。

1.2 练习并熟悉基于 MIPS 的汇编语言的基本操作和数据读入

2. 练习和辨析三种寻址方式：直接寻址、间接寻址、基线寻址。熟悉算数计算和内存的装载和储存。

二、实验内容

1. 在以下两种情况下计算并输出(a+b*c)/d 的商和余数,同时输出自己的学号和姓名。
   1.1. a,b,c,d 的值在 source file 中给定 (a,b,c,d 为整型数)
   1.2. a,b,c,d 的值由用户给定（a,b,c,d 为整型数）
2. 将字符串“abc”改为大写样式，在屏幕上打印输出。

三、实验步骤（阐述代码思路或操作步骤）

详细实验步骤及实验思路（红色）均在代码的注释内给出

以及每个操作的目的、结果和寄存器的变化（蓝色）

1.1

```
.data
    # define the variables and notes that will be used in memory
    a: .word 1
    b: .word 2
    c: .word 3
    d: .word 4
    n: .asciiz"\n"
    str1: .asciiz "The Quotient is: "
    str2: .asciiz " ,reminder is: "
    sid: .asciiz "SID: 11710823"
    name: .asciiz "Name: Yanbin Wei"
    note: .asciiz "Set a=1,b=2,c=3,d=4, calculate (a+b*c)/d."

.text
    main:
    #load a,b,c,d from memory to the registers
    lw $t0 a  #t0: 1
    lw $t1 b  #t1: 2
    lw $t2 c  #t1: 3
    lw $t3 d  #t2: 4
```

```
#print the note: value and question
li $v0 4            #v0: 4
la $a0 note         #a0: the adress of note
syscall             #print the notation
la $a0 n            #a0: the adress of n
syscall             #change a line

#compute the formula
mul $t4 $t1 $t2     #t4: 6
add $t5 $t0 $t4     #t5: 7
divu $t5 $t3        #lo: 1 hi: 3
mflo $t6            #t6: 1
mfhi $t7            #t7：  3

#print the quotient
li $v0 4        #v0: 4
la $a0 str1     #a0: address of str1
syscall         #print note str1
li $v0 1        #v0: 1
move $a0 $t6 #a0: 1
syscall         #print the quotient

#print the reminder
li $v0 4        #v0: 4
la $a0 str2     #a0: address of str2
syscall         #print note str2
li $v0 1        #v0: 1
move $a0 $t7 #a0: 3
syscall         #print reminder
li $v0 4        #v0: 4
la $a0 n        #a0: address of n
syscall         #change a line

#print my student sid
li $v0 4        #v0: 4
la $a0 sid      #a0:address of sid
syscall         #print my student id
la $a0 n        #a0:address of n
syscall         #change a line

#print my name
la $a0 name   #a0:address of name
```

```
	syscall			#print my name
	la $a0 n		#a0: the address of n
	syscall			#change a line

	#exit the program
	li $v0 10		#v0: 10
	syscall			#exit
```

1.2

```
.data
	# define the variables and notes that will be used in memory
	n: .asciiz"\n"
	str1: .asciiz "The Quotient is: "
	str2: .asciiz " ,reminder is: "
	sid: .asciiz "SID: 11710823"
	name: .asciiz "Name: Yanbin Wei"
	note: .asciiz "Set a=1,b=2,c=3,d=4, calculate (a+b*c)/d."

.text
	main:
	#read a
	li $v0 5		#v0: 5
	syscall			#read a
	move $t0 $v0		#t0: a

	#read b
	li $v0 5
	syscall			#read b
	move $t1 $v0		#t1: b

	#read c
	li $v0 5
	syscall			#read c
	move $t2 $v0		#t2: c

	#read d
	li $v0 5
	syscall			#read d
	move $t3 $v0		#t3: d

	#print the note: value and question
```

```
li $v0 4            #v0: 4
la $a0 note         #a0: the adress of note
syscall             #print the notation
la $a0 n            #a0: the adress of n
syscall             #change a line

#compute the formula
mul $t4 $t1 $t2     #t4: 6
add $t5 $t0 $t4     #t5: 7
divu $t5 $t3        #lo: 1 hi: 3
mflo $t6            #t6: 1
mfhi $t7            #t7：  3

#print the quotient
li $v0 4        #v0: 4
la $a0 str1     #a0: address of str1
syscall         #print note str1
li $v0 1        #v0: 1
move $a0 $t6 #a0: 1
syscall         #print the quotient

#print the reminder
li $v0 4        #v0: 4
la $a0 str2     #a0: address of str2
syscall         #print note str2
li $v0 1        #v0: 1
move $a0 $t7 #a0: 3
syscall         #print reminder
li $v0 4        #v0: 4
la $a0 n        #a0: address of n
syscall         #change a line

#print my student sid
li $v0 4        #v0: 4
la $a0 sid      #a0:address of sid
syscall         #print my student id
la $a0 n        #a0:address of n
syscall         #change a line

#print my name
la $a0 name  #a0:address of name
syscall         #print my name
```

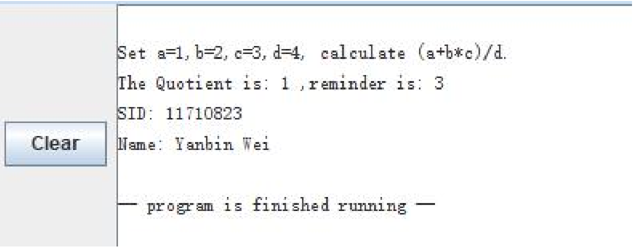
```
        la $a0 n          #a0: the address of n
        syscall           #change a line

        #exit the program
        li $v0 10         #v0: 10
        syscall           #exit
```

```
2. .data
        str: .asciiz "a","b","c"
.text
        #load "a","b","c" to regester
        li $v0 4          #v0 4
        la $a0 str        #a0:address of str
        lb $t0 ($a0)      #t0:ascii of a
        lb $t1 2($a0)     #t1:ascii of b
        lb $t2 4($a0)     #t2:ascii of c

        #use sub to decrease ascii
        subi $t0,$t0,32       #t0: ascii of A
        subi $t1,$t1,32       #t1: ascii of B
        subi $t2,$t2,32       #t2: ascii of C

        #store the result return to the memory
        sb $t0 ($a0)
        sb $t1 2($a0)
        sb $t2 4($a0)
        syscall           #print A
        addi $a0,$a0,2        #a0: address of str+2
        syscall           #print B
        addi $a0,$a0,2        #a0: address of str+4
        syscall           #print C

        #exit
        li $v0 10
        syscall           #exit
```

四、实验结果（截图并配以适当的文字说明）

1.1

运行结果如下，输出包含题目要求的答案和所设定好的提示信息

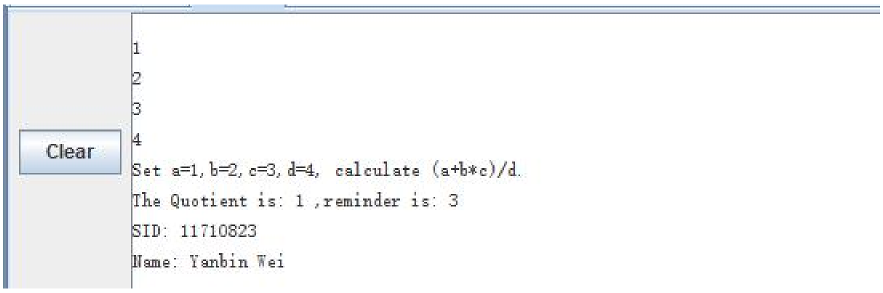


```
Set a=1,b=2,c=3,d=4, calculate (a+b*c)/d.
The Quotient is: 1 ,reminder is: 3
SID: 11710823
Name: Yanbin Wei

— program is finished running —
```

1.2

运行结果如下，前四行是输入的 a,b,c,d 的值

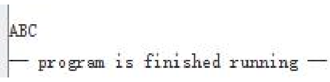
输出包含题目要求的答案和提示信息



```
1
2
3
4
Set a=1,b=2,c=3,d=4, calculate (a+b*c)/d.
The Quotient is: 1 ,reminder is: 3
SID: 11710823
Name: Yanbin Wei
```

2

运行结果如下，输出按题目要求，将“abc“”转化为了“ABC”



```
ABC
— program is finished running —
```

五、实验分析（遇到的问题以及解决方案）

问题 1：输出并非预测的值而是地址

解决：检查发现对 load 操作的使用有误，此处应当使用 lw 间接取值而非 la 的直接取址

问题 2：实验报告 word 排版十分复杂

解决：使用高亮对代码注释进行分类标记

六、实验小结与体会

练习并熟悉了汇编语言基于寄存器的几项基本操作（算数计算、数据装载、系统命令（输入输出），变量定义和特定寄存器的使用等）

辨析了三种不同的寻址方式

体会到了汇编语言相对于高级语言的基础性和底层性

（请写完后另存为 PDF 再提交，其他格式包括 word 一律扣分）