实验内容 MIPS 汇编程序设计

——电信提高 2101 班杨筠松 U202115980

一、实验任务

采用 MIPS 汇编程序实现以下功能：

1）在数据段定义两个 int 型变量 a、b，

2）在数据段定义一个 int 型数组 c[40]，不进行初始化。

3）通过系统功能调用从键盘输入 a，b 的值（不大于 20）

4）采用 MIPS 汇编指令实现 c[a + b] = a * b;

5）通过系统功能调用分别显示 c[a + b]的所在的存储地址和值

6）指出程序运行结束后 a，b，c[a + b]所在的数据段存储位置以及取值，验证程序功能的正确性

二、实验目的

1. 掌握 MIPS 汇编指令
2. 熟悉 MIPS 汇编语言程序结构
3. 掌握 C 语言语句 MIPS 汇编语言指令实现方案，了解 C 语言编译原理
4. 掌握 MIPS 汇编语言程序数据段、指令段内存映像
5. 熟练掌握使用 MIPS 汇编语言模拟器 MARS 调试汇编语言程序
6. 掌握利用 MIPS 汇编语言模拟器获取 MIPS 汇编语言源程序对应机器指令

三、实验环境

1. Window11 操作系统
2. 编辑工具：Mars4.5
3. MIPS 模拟器：Mars4.5
4. 模拟器运行环境：Java19 SE

四、汇编语言设计思路

1)变量定义

实验中要求在数据段定义两个 int 类型的变量 a,b，并且在数据段定义一个 int 类型的数据 c[40]，不初始化。通过 help 手册了解到通过关键字.data 字段声明数据定义段，利用关键字.space 为变量设置字节数。

```
1  .data
2  a:.space 4
3  b:.space 4
4  c:.space 160
```

2)通过系统调用实现变量 a,b 的读入

通过 help 手册了解到通过调用系统函数可以实现变量 a,b 的读取，如下图所示

```
17  la $s0, a
18  li $v0, 5
19  syscall
20  sw $v0, 0($s0)
21  # acquire the value of the variable 'a'
```

```
28  la $s0, b
29  li $v0, 5
30  syscall
31  sw $v0, 0($s0)
32  # acquire the value of the variable 'b'
```

3）通过 Basic instruction 的组合实现语句 c[a + b] = a * b

先进行 a + b 的实现，将变量 a，b 分别载入到寄存器$t0，$t1 中，随之利用 add 命令实现 a + b 并将结果存储在$t0 中，同时利用 mult 指令实现 a * b 并且将低 32 位存储在$t1 中。接下来实现对地址 c[a + b]的存储，即将 c 地址同 4 * (a + b)的结果加和得到 c[a + b]地址，使用 sw 指令进行存储即可。

```
34  la $s0, a
35  la $s1, b
36  lw $t0, 0($s0)
37  lw $t1, 0($s1)
38  add $t2, $t0, $t1
39  li $v0, 4
40  la $a0, straplusb
41  syscall
42  li $v0, 1
43  add $a0, $t2, 0
44  syscall
45  # output the value of 'a + b'
46
47  add $t0, $t2, 0                    ## at this time, $t0 is a + b
```

```
53  la $s0, a
54  la $s1, b
55  lw $t1, 0($s0)
56  lw $t2, 0($s1)
57  mult $t1, $t2
58  mflo $t1                           ## at this time, $t1 = a  * b
```

```
67  la $s2, c
68  sll $t3, $t0, 2
69  add $t3, $t3, $s2
70  sw $t1, 0($t3)                                  # finish the storage
```

4)打印相关变量和地址

通过 help 手册可知可以通过 syscall 实现打印功能

```
76  li $v0, 4
77  la $a0, straddress
78  syscall
79  li $v0, 1
80  add $a0, $t3, 0
81  syscall
```

```
li $v0, 1
lw $t2, 0($t3)
add $a0, $t2, 0
syscall
```

五、实现过程

完整的源代码和注释如下所示

```
.data
a:.space 4
b:.space 4
c:.space 160
stra:.asciiz "Please input the value of a:\n"
strb:.asciiz "Please input the value of b:\n"
straplusb:.asciiz "the value of a + b = "
stratimesb:.asciiz "the value of a * b = "
straddress:.asciiz "the address of c[a+b] is  "
another:.asciiz "\n"

.text
li $v0, 4
la $a0, stra
syscall
# output the hint of 'a' variable
la $s0, a
li $v0, 5
syscall
sw $v0, 0($s0)
# acquire the value of the variable 'a'


li $v0, 4
la $a0, strb
syscall
# output the hint of 'b' variable
la $s0, b
li $v0, 5
syscall
sw $v0, 0($s0)
# acquire the value of the variable 'b'

la $s0, a
la $s1, b
lw $t0, 0($s0)
lw $t1, 0($s1)
add $t2, $t0, $t1
li $v0, 4
la $a0, straplusb
syscall
li $v0, 1
add $a0, $t2, 0
syscall
# output the value of 'a + b'

add $t0, $t2, 0                         ## at this time, $t0 is a + b

li $v0, 4
```

```
la $a0, another
syscall

la $s0, a
la $s1, b
lw $t1, 0($s0)
lw $t2, 0($s1)
mult $t1, $t2
mflo $t1                                    ## at this time, $t1 = a  * b
li $v0, 4
la $a0, stratimesb
syscall
li $v0, 1
add $a0, $t1, 0
syscall
# output the value of 'a*b'

la $s2, c
sll $t3, $t0, 2
add $t3, $t3, $s2
sw $t1, 0($t3)                                  # finish the storage

li $v0, 4
la $a0, another
syscall

li $v0, 4
la $a0, straddress
syscall
li $v0, 1
add $a0, $t3, 0
syscall

li $v0, 4
la $a0, another
syscall

li $v0, 1
lw $t2, 0($t3)
add $a0, $t2, 0
syscall

li $v0, 10
syscall
```

六、实验结果

输入变量 a 的值为 15 变量 b 的值为 18 期望输出如下：

$$a+b=33\,,a*b=270$$

实际输出结果如下所示：

```
Please input the value of a:
15
Please input the value of b:
18
the value of a + b = 33
the value of a * b = 270
the address of c[a+b] is  8332
270
-- program is finished running --
```

查看栈区有：

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
8192	15	18	0	0	0	0	0	0
8224	0	0	0	0	0	0	0	0
8256	0	0	0	0	0	0	0	0
8288	0	0	0	0	0	0	0	0
8320	0	0	0	270	0	0	0	0
8352	0	0	1634036816	1763730803	1953853550	1701344288	1818326560	1864394101
8384	979443814	1817182218	1702060389	1886284064	1948284021	1981834600	1702194273	543584032
8416	670306	543516788	1970037110	1718558821	723542304	1025532448	1752432672	1635131493
8448	543520108	1629513327	1646275104	2112800	543516788	1919181921	544437093	1663067759
8480	1647010139	1936269405	167780384	0	0	0	0	0
8512	0	0	0	0	0	0	0	0
8544	0	0	0	0	0	0	0	0
8576	0	0	0	0	0	0	0	0
8608	0	0	0	0	0	0	0	0

对照相应地址可找到期望值，说明实验功能已完全完成

七、实验总结

本次实验是使用 Mars 作为 MIPS 模拟器实现 MIPS 汇编语言编程，收获颇丰，将课本上枯燥的知识完成了内化吸收用于了实践当中，其中比较有趣的是断点调试部分，令人耳目一新，之前一直使用 gdb 来调试 C 和 C++程序，经过这次实验终于找到了原型。实验内容虽然较为简单，但是对本人理解程序如何执行以及模拟器使用有极大的帮助。