LAB #03 Report

Name:黄家睿 ID number:202283890036 Major:IOT

Task1：

1. Write a program to ask the user to enter two integers A and B and then display the result of computing the expression: A + 2B - 5.

Solution：

Code part：

Edit | Execute

Lab03.asm

```
1  .data
2  str1: .asciiz "Enter an interger value: "
3  str2: .asciiz "The result is: "
4
5  .globl main
6  .text
7  main:
8          #input the first interge into $s0
9          li $v0,4         #service code for print string
10         la $a0,str1      #service code for load address str1 to $a0
11         syscall
12         li $v0,5         #service code for read interge
13         syscall
14         move $s0,$v0     #save input integer into $s0
15                                                          syscall
16         #input the second integer into $s1
17         li $v0,4         #service code foe print string
18         la $a0,str1      #load the str1 into $a0
19         syscall
20         li $v0, 5        #service code for read integer
21         syscall
22         move $s1,$v0
23
24         #calculate A+2B-5
25         add $s2,$s1,$0
26         add $s1,$s1,$s2 #double the second integer
27         add $s0,$s0,$s1 #A+2B
28         subi $s0,$s0,5  #A+2B-5
29
30
```

```
31          #output the result
32          li $v0, 4           #service code for print string
33          la $a0, str2        #load str2 address into $a0
34          syscall
35          li $v0, 1           #service code for print integer
36          move $a0, $s0
37          syscall
38
39          li $v0, 10
40          syscall
```

Output part：

```
Enter an interger value: 5
Enter an interger value: 5
The result is: 10
— program is finished running —
```

Task2

2. Assume that $s1 = 0x12345678 and $s2 = 0xffff9a00. Determine the content of registers $s3 to $s6 after executing the following instructions:

```
and $s3,$s1,$s2 # $s3 =
or  $s4,$s1,$s2 # $s4 =
xor $s5,$s1,$s2 # $s5 =
nor $s6,$s1,$s2 # $s6 =
```

Write a program to execute these instructions and verify the content of registers $s3 to $s6.

Solution：

Code part:

```
.data
str1: .asciiz "$s3= "
str2: .asciiz "$s4= "
str3: .asciiz "$s5= "
str4: .asciiz "$s6= "
str5: .asciiz "\n"

.globl main
.text
main:
        li $s1,0x12345678
        li $s2,0xffff9a00

        and $s3,$s1,$s2 #$s3=
        or $s4,$s1,$s2  #$s4=
        xor $s5,$s1,$s2 #$s5=
        nor $s6,$s1,$s2 #$s4=

        li $v0,4        #service code for print strin
        la $a0,str1     #load str1 into $a0
        syscall
        move $a0,$s3
        li $v0,34       #service code for print integer
        syscall
        li $v0,4
        la $a0,str5
        syscall

        li $v0,4        #service code for print strin
        la $a0,str2     #load str1 into $a0
        syscall
        move $a0,$s4
        li $v0,34       #service code for print integer
        syscall
        li $v0,4
        la $a0,str5
        syscall

        li $v0,4        #service code for print strin
        la $a0,str3     #load str1 into $a0
        syscall
        move $a0,$s5
        li $v0,34       #service code for print integer
        syscall
        li $v0,4
        la $a0,str5
        syscall

        li $v0,4        #service code for print strin
        la $a0,str4     #load str1 into $a0
        syscall
        move $a0,$s6
        li $v0,34       #service code for print integer
        syscall
        li $v0,4
        la $a0,str5
        syscall

        li $v0,10
        syscall
```

Output part:

```
$s3= 0x12341200
$s4= 0xffffde78
$s5= 0xedcbcc78
$s6= 0x00002187
```

Task3:

3. Assume that $s1 = 0x87654321. Determine the content of registers $s2 to $s4 after executing the following instructions:

```
sll $s2,$s1, 16 # $s2 =
srl $s3,$s1, 8  # $s3 =
sra $s4,$s1, 12 # $s4 =
```

Write a program to execute these instructions and verify the content of registers $s2 to $s4.

Solution:

Code part:

```
.data
str1: .asciiz "$s2= "
str2: .asciiz "$s3= "
str3: .asciiz "$s4= "
str4: .asciiz "\n "

.globl main
.text
main:
        li $s1,0x87654321       #intilize $s1 into 0x87654321
        sll $s2,$s1,16
        srl $s3,$s1,8
        sra $s4,$s1,12

        li $v0,4        #service code for print srting
        la $a0,str1
        syscall
        move $a0,$s2
        li $v0,34
        syscall
        li $v0,4
        la $a0,str4
        syscall
```

```
24
25          li $v0,4        #service code for print srting
26          la $a0,str2
27          syscall
28          move $a0,$s3
29          li $v0,34
30          syscall
31          li $v0,4
32          la $a0,str4
33          syscall
34
35          li $v0,4        #service code for print srting
36          la $a0,str3
37          syscall
38          move $a0,$s4
39          li $v0,34
40          syscall
41          li $v0,4
42          la $a0,str4
43          syscall
44
45          li $v0,10
46          syscall
```

Output part:

```
$s2= 0x43210000
$s3= 0x00876543
$s4= 0xfff87654
```

Task4:

4. Write a program that asks the user to enter an alphabetic character (either lower or upper case) and change the case of the character from lower to upper and from upper to lower and display it. (Hint: ASCII code of 'a' = 0x61 and 'A' = 0x41. So, in case of 'a', 0x61 – 0x20 will give us upper case letter 'A'. For the ASCII values, you can refer to Fig. 5)

Solution:

Code part:

```
.data
prompt: .asciiz "Enter an alphabetic character: "
newline: .asciiz "\n"
resultChar: .asciiz "Converted charater: "
error_msg: .asciiz "Invalid input. Please enter an alphabetic character.\n"


.text
main:
    # Prompt the user for input
    li $v0, 4       # Print string
    la $a0, prompt
    syscall

    # Read a character from user input
    li $v0, 12      # Read character
    syscall
    move $t0, $v0   # Store the input character
    li $v0,4
    la $a0,newline
    syscall

    #check if lower case
    li $t1,'a'
    li $t2,'z'
    bgt $t0,$t2,Error_case
    blt $t0,$t1,upper_case
```

```
38          #convert lower case
39          add $t0,$t0,$t3
40  print_result:
41          li $v0,4        #service code for print string
42          la $a0,resultChar
43          syscall
44          li $v0,11
45          move $a0,$t0
46          syscall
47          j exit
48  Error_case:
49          li $v0,4        #service code for print string
50          la $a0,error_msg
51          syscall
52
53  exit:
54          li $v0,10
55          syscall
```

Output part：

```
Enter an alphabetic character: u
Converted charater: U
-- program is finished running --
```

Task5：

5. Write a program that asks the user to enter an unsigned number and read it. Then swap the bits at odd positions with those at even positions and display the resulting number. For example, if the user enters the number 9, which has binary representation of 1001, then bit 0 is swapped with bit 1, and bit 2 is swapped with bit 3, resulting in the binary number 0110. Thus, the program should display 6.

Solution ：

Code part：

Lab03_task5.asm

```
1   .data
2   str1: .asciiz "Please enter an integer value"
3   str2: .asciiz "The result is: "
4
5
6   .text
7   main:
8           #input the integer
9           li $v0,4
10          la $a0,str1
11          syscall
12          li $v0,5
13          syscall
14          move $t0,$v0
15
16          li $t1,0xAAAAAAAA
17          li $t2,0X55555555
18          and $t3,$t0,$t1
19          and $t4,$t2,$t0
20          srl $t3,$t3,1
21          sll $t4,$t4,1
22          or $t0,$t3,$t4
23
24          li $v0,4
25          la $a0,str2
26          syscall
27          li $v0,1
28          move $a0,$t0
29          syscall
30
```

Output part：

```
Please enter an integer value9
The result is: 6
— program is finished running —
```