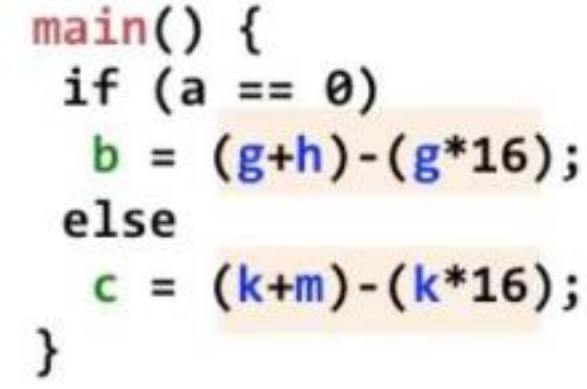
DHA Suffa University
Department of Computer Science
Computer Organization & Assembly Language
Spring 2021
Lab # 8 (Procedures in MIPS)

Objective:

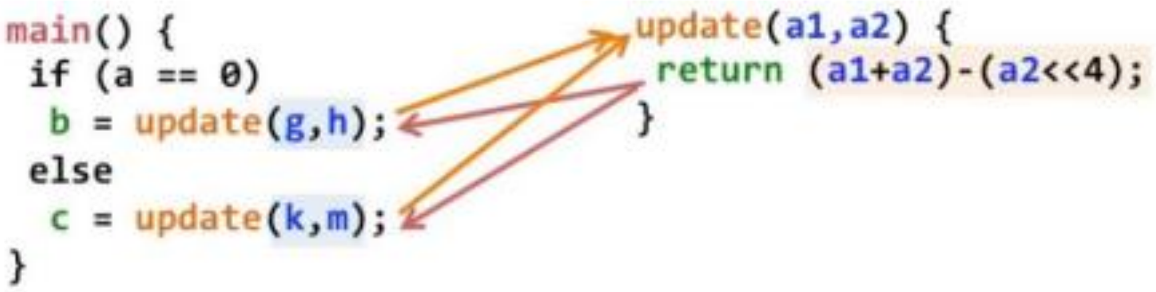
To deal with Procedures in MIPS.

Understand Procedures in general:



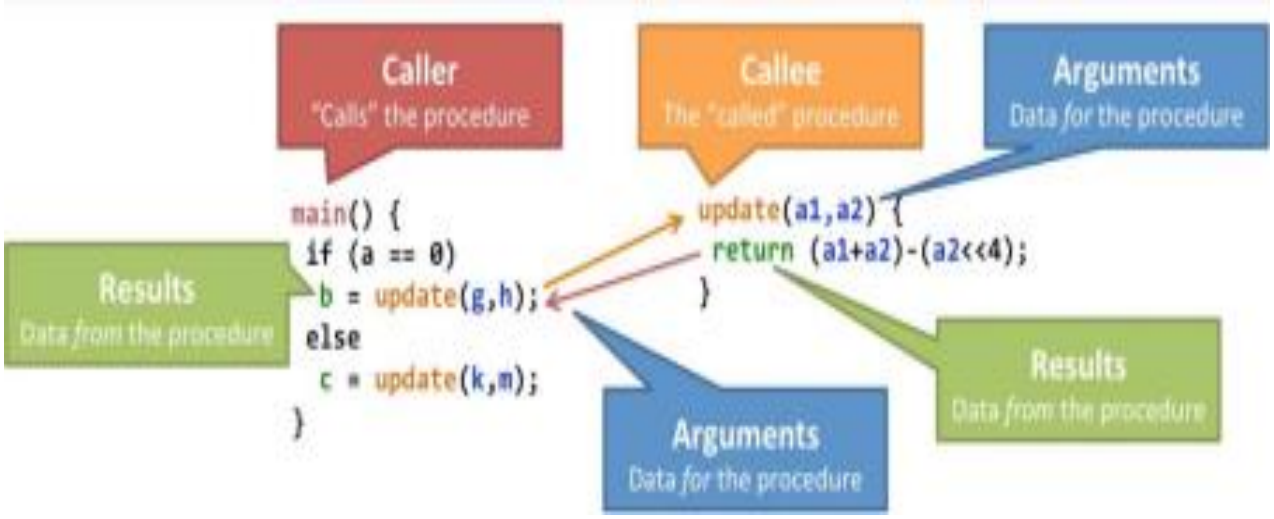
```
main() {
 if (a == 0)
  b = (g+h)-(g*16);
 else
  c = (k+m)-(k*16);
}
```

Can this code will converted into Procedures??



```
main() {                    update(a1,a2) {
 if (a == 0)                 return (a1+a2)-(a2<<4);
  b = update(g,h);          }
 else
  c = update(k,m);
}
```

Procedure call Terminology:



Procedures in MIPS:

Following are some conventions used in procedure calls:
$a0-$a3 for arguments 1-4 of a procedure.

$v0-$v1 for results of a procedure.

$s0-$s7 are the saved registers, these registers should be unchanged after a procedure call. **$t0-$t9** are temporary registers, and are not necessarily preserved across procedure calls. **Following instructions are normally used in Procedure calls:**

jal *label*

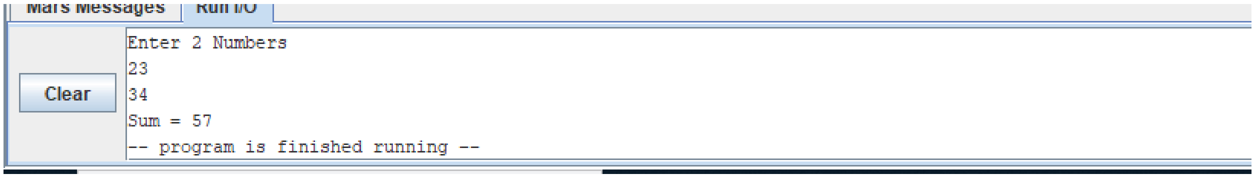
It Copies the address of the next instruction **($pc+4**) into the register **$ra** and then jumps to the address label.

jr *$register:* It jumps to the address in $**register**. It's most common use is **jr $ra.**

Example 01:

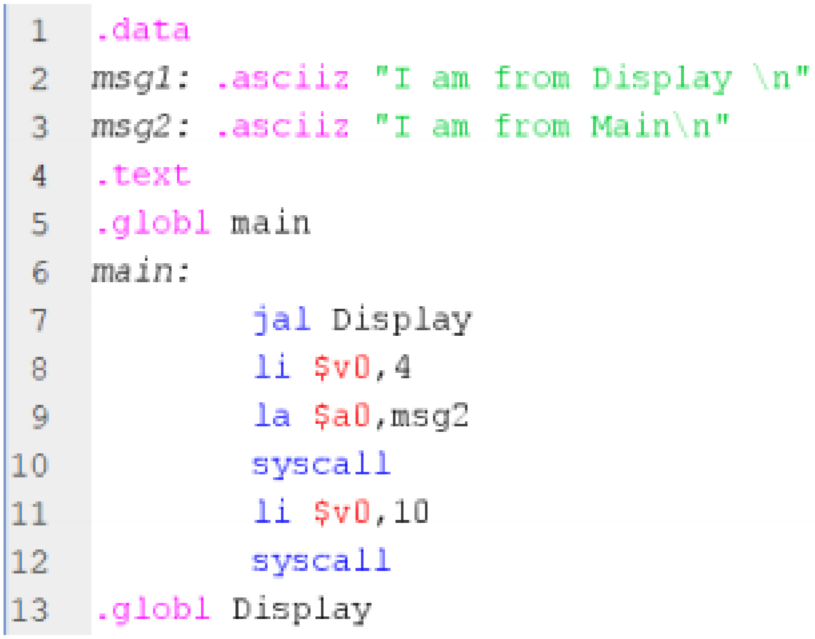
```
.data
string1:.asciiz "Enter 2 Numbers\n"
string2:.asciiz "Sum = "
.text
jal Sum
li $v0, 10
syscall

Sum:
li $v0, 4
la $a0, string1
syscall

li $v0, 5
syscall
move $s0, $v0

li $v0, 5
syscall
move $s1, $v0

add $t0, $s1, $s0
```

```
li $v0,4

la $a0, string2

syscall


li $v0, 1

move $a0, $t0

syscall

jr $ra
```

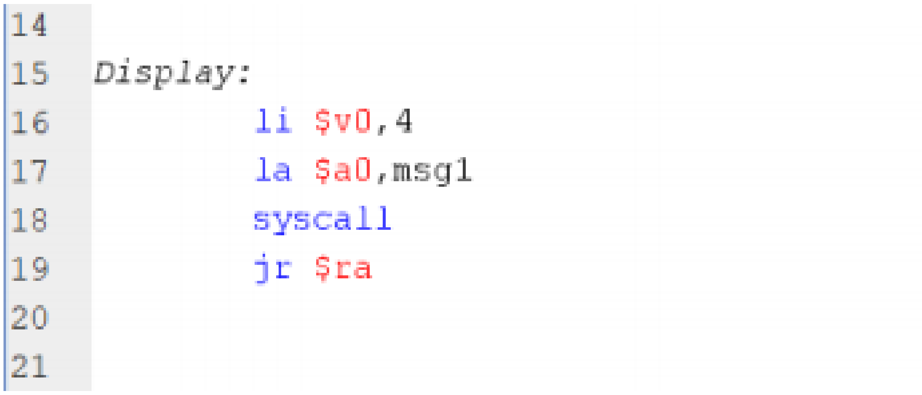


```
Enter 2 Numbers
23
34
Sum = 57
-- program is finished running --
```

Example 02:



```
.data
msg1: .asciiz "I am from Display \n"
msg2: .asciiz "I am from Main\n"
.text
.globl main
main:
        jal Display
        li $v0,4
        la $a0,msg2
        syscall
        li $v0,10
        syscall
.globl Display
```



```
Display:
        li $v0,4
        la $a0,msg1
        syscall
        jr $ra
```

Example 03:

```
.data

string1:.asciiz "Enter 2 Numbers\n"

string2:.asciiz "Greater = "

.text


jal Greater


Exit:

li $v0, 10

syscall


Greater:

li $v0, 4

la $a0, string1

syscall
```

```
li $v0, 5
syscall
move $s0, $v0

li $v0, 5
syscall
move $s1, $v0

bgt $s0, $s1, Great

li $v0,4
la $a0, string2
syscall

li $v0, 1
move $a0, $s1
syscall


jr $ra
```

```
Great:

li $v0,4

la $a0, string2

syscall


li $v0, 1

move $a0, $s0

syscall


j Exit
```

LAB TASK

Write a program to input two numbers, containing 5 procedures to calculate the sum, product, Difference, quotient and remainder.

LAB ASSIGNMENT

You are required to write a procedure which takes a string as an input and returns the largest word length in the string.

Sample Input:
Computer Organization and Assembly Language
Sample Output:
12