

Lab Report

ECPE 170 – Computer Systems and Networks – Spring 2021

Name: Kaung Khant Pyae Sone

Lab Topic: MIPS Assembly Programming (Basic) (Lab #: 10)

Question #1:

Take two screenshots of the MIPS register panel: one before your program runs, and one after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Answer:

Before:

```
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

After:

```
R8  [t0] = 30
R9  [t1] = 5
R10 [t2] = 7
R11 [t3] = 8
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 15
R17 [s1] = 10
R18 [s2] = 7
R19 [s3] = 2
R20 [s4] = 18
R21 [s5] = -3
R22 [s6] = 37
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

Question #2:

Take two screenshots of the MIPS memory panel (data tab): one before your program runs, and one after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. **In the after-execution capture, circle the memory location (not register) that contains the final calculated value of Z.**

Answer:

Before:

```
User data segment [10000000]..[10040000]
[10000000]..[1003ffff]  00000000
```

After:

```
User data segment [10000000]..[10040000]
```

```
[10000000]..[1000ffff] 00000000
```

[10010000] 0000000037 0000000000 0000000000 0000000000 000

```
[10010010]..[1003ffff] 00000000
```

Question #3:

Take two screenshots of the MIPS register panel: one before your program runs, and one after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Answer:

Before:

```
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

After:

```
R8  [t0] = 2
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 10
R17 [s1] = 15
R18 [s2] = 6
R19 [s3] = 0
R20 [s4] = 268501004
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

Question #4:

Take two screenshots of the MIPS memory panel (data tab): one before your program runs, and one after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. In the after-execution capture, **circle the memory location (not register) that contains the final calculated value of Z**

Answer:

Before:

```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 0000000010 0000000015 0000000006 0000000000 . . . . .
[10010010]..[1003ffff] 00000000
```

After:

```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 0000000010 0000000015 0000000006 0000000000 . . . . .
[10010010]..[1003ffff] 00000000
```

Question #5:

Take a screenshot of the MIPS register panel after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Answer:

Before:

```
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

After:

```
R8  [t0] = 100
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 78
R18 [s2] = 268500992
R19 [s3] = 268500996
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

Question #6:

Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. **Circle the memory location (not register) that contains the final calculated values of I and Z.**

Answer:

Before:

```
User data segment [10000000]..[10040000]
[10000000]..[10010003] 00000000
[10010004] 0000000002 0000000000 0000000000 . . . . .
[10010010]..[1003ffff] 00000000
```

After:

```

      I
User data segment [10000000]..[10040000]
[10000000]..[10010003] 00000000
Z [10010004] 0000000078 0000000000 0000000000 N . . . . .
[10010010]..[1003ffff] 00000000
```

Question #7:

Take a screenshot of the MIPS register panel after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Answer:

Before:

```
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

After:

```
R8  [t0] = 4
R9  [t1] = 0
R10 [t2] = 268501012
R11 [t3] = 26
R12 [t4] = 5
R13 [t5] = 17
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 268500992
R17 [s1] = 268501012
R18 [s2] = 12
R19 [s3] = -1
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

Question #8:

Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. **Circle the final values of array A.**

Answer:

Before:

```
User data segment [10000000]..[10040000]
[10000000]..[1003ffff] 00000000
```

After:

```
User data segment [10000000]..[10040000]
[10000000]..[10010013] 00000000
[10010014] 0000000026 0000000028 0000000030
[10010020] 0000000032 0000000034 0000000000 0000000000
[10010030]..[1003ffff] 00000000
```

Question #9:

Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Hex mode (right-click), since Decimal mode will not allow us to distinguish between bytes.

Circle two things: the final value of the pointer 'result' in memory, and the corresponding location that result points to. Does that location in memory contain the ASCII code for the character 'e'? (If not, you had better check your work!)

Answer:

The screenshot shows the MIPS memory panel with the 'Data' tab selected. The 'FP Regs' and 'nt Regs [16]' tabs are also visible. The 'Int Regs [16]' section shows the following register values:

Register	Value
PC	4000dc
EPC	0
Cause	0
BadVAddr	0
Status	3000ff10
HI	0
LO	0
R0 [r0]	0
R1 [at]	10010000
R2 [v0]	a
R3 [v1]	0
R4 [a0]	65
R5 [a1]	7ffff9d0
R6 [a2]	7ffff9d8
R7 [a3]	0
R8 [t0]	65
R9 [t1]	10010101
R10 [t2]	65
R11 [t3]	65
R12 [t4]	0
R13 [t5]	0
R14 [t6]	0
R15 [t7]	0
R16 [s0]	1
R17 [s1]	10010000
R18 [s2]	10010101

The 'Data' tab shows the 'User data segment [10000000]..[10040000]' and the 'User Stack [7ffff9cc]..[80000000]'. The 'User data segment' is shown in decimal mode, and the 'User Stack' is shown in hex mode. The 'User data segment' contains the following data:

Address	Value
[10000000]..[100100ff]	00000000
[10010100]	7435574
[10010110]	72694600
[10010120]	72646461
[10010130]	6e696863
[10010140]	00207369
[10010150]	000a646e
[10010160]..[1003ffff]	00000000

The 'User Stack' is shown in hex mode, and the 'Data' tab is selected. The 'Data' tab shows the 'User data segment [10000000]..[10040000]' and the 'User Stack [7ffff9cc]..[80000000]'. The 'User data segment' is shown in decimal mode, and the 'User Stack' is shown in hex mode. The 'User data segment' contains the following data:

Address	Value
[10000000]..[100100ff]	00000000
[10010100]	7435574
[10010110]	72694600
[10010120]	72646461
[10010130]	6e696863
[10010140]	00207369
[10010150]	000a646e
[10010160]..[1003ffff]	00000000

The 'User Stack' is shown in hex mode, and the 'Data' tab is selected. The 'Data' tab shows the 'User data segment [10000000]..[10040000]' and the 'User Stack [7ffff9cc]..[80000000]'. The 'User data segment' is shown in decimal mode, and the 'User Stack' is shown in hex mode. The 'User data segment' contains the following data:

Address	Value
[10000000]..[100100ff]	00000000
[10010100]	7435574
[10010110]	72694600
[10010120]	72646461
[10010130]	6e696863
[10010140]	00207369
[10010150]	000a646e
[10010160]..[1003ffff]	00000000

(1) How would you suggest improving this lab in future semesters?

Variations on load and store also exist for smaller data sizes:

- 16-bit halfword: `lh` and `sh`
- 8-bit byte: `lb` and `sb`

This section on the MIPS Instruction Set page should be placed inside a table so that it is more visible.