

## Lab 0

### 1.

#### 1.1

#### Text :

```

User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29)           ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4        ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4        ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2          ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2        ; 187: addu $a2 $a2 $v0
[00400014] 0c000000 jal 0x00000000 [main]   ; 188: jal main
[00400018] 00000000 nop                   ; 189: nop
[0040001c] 3402000a ori $2, $0, 10        ; 191: li $v0 10
[00400020] 0000000c syscall                ; 192: syscall # syscall 10 (exit)

```

อธิบาย : จากภาพเป็นโค้ดในส่วนของ text ซึ่งเป็น Default ของโปรแกรม QtSpim

#### Data :

```

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

```

อธิบาย : จากภาพเป็นการจองพื้นที่ใน Memory โดยถูกกำหนดให้ค่าเริ่มต้น(initial) ใน Memory นั้นมีค่าเป็น 00000000 โดยจากภาพ Address ใน Memory ที่เริ่มต้นที่ 10000000 ถึง 1000000f แล้วจะเริ่มทำต่อที่ Address 10000010 จนถึง 1000001f เป็นเช่นนี้ไปเรื่อยๆ จนถึง Address ที่ 1003ffff

นางสาวจิรัชฌา วิทยาชีวะ 6010502519

นางสาวจาวรี ทองเจริญ 6010504660

### Int Regs :

อธิบาย : จากภาพเป็นการกำหนดค่าใน Registers โดยแสดงเป็นจำนวน Integer (R0-R31) และ Registers พิเศษต่างๆ โดยถูกกำหนดมาให้ค่าเริ่มต้น(initial) ในส่วนของ R8 ถึง R15 เป็น 0 เนื่องจาก R8 ถึง R15 นั้น เป็นค่า Temporary Registers ซึ่งเราจะใช้จองพื้นที่ใน Registers ส่วนนี้เป็นส่วนใหญ่

Int Regs [10]	
PC	= 0
EPC	= 0
Cause	= 0
BadVAddr	= 0
Status	= 805371664
HI	= 0
LO	= 0
R0 [x0]	= 0
R1 [at]	= 0
R2 [v0]	= 0
R3 [v1]	= 0
R4 [a0]	= 1
R5 [a1]	= 2147483148
R6 [a2]	= 2147483156
R7 [a3]	= 0
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
R26 [k0]	= 0
R27 [k1]	= 0
R28 [gp]	= 268468224
R29 [sp]	= 2147483144
R30 [s8]	= 0
R31 [ra]	= 0

## 1.2

## Text :

```

[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
[00400024] 3c081001 lui $8, 4097 [my_array] ; 10: la $8, my_array # move address of my_array (&my_array) into $8
[00400028] 8d090000 lw $9, 0($8) ; 11: lw $9, 0($8) # load my_array[0] into $9
[0040002c] 8d0a0004 lw $10, 4($8) ; 12: lw $10, 4($8) # load my_array[1] into $10
[00400030] 012a5820 add $11, $9, $10 ; 13: add $11, $9, $10 # add the two numbers into $11
[00400034] 8d090008 lw $9, 8($8) ; 14: lw $9, 8($8) # load my_array[2] into $9
[00400038] 01695820 add $11, $11, $9 ; 15: add $11, $11, $9 # add the number in $11 to it and accumulate to $11
[0040003c] ad0b000c sw $11, 12($8) ; 16: sw $11, 12($t0) # store the added result to my_array[3]
[00400040] 3402000a ori $2, $0, 10 ; 20: li $v0, 10
[00400044] 0000000c syscall ; 21: syscall

```

อธิบาย : จากภาพเป็นโค้ดในส่วนของ text เมื่อโหลดโปรแกรม sample1.asm แต่ยังไม่ได้อินโปรแกรมจะแสดงโค้ดที่ได้จาก sample1.asm ในส่วนของ text

## Data :

```

User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 0000000100 -200 0000000500 0000000000 d . . . 8 . . . . .
[10010010]..[1003ffff] 00000000

```

อธิบาย : จากภาพเป็นการจองพื้นที่ใน Memory โดยถูกกำหนดให้ค่าเริ่มต้น(initial) ใน Memory นั้นมีค่าเป็น 00000000 โดยจากภาพ Address ใน Memory ที่เริ่มต้นที่ 10000000 ถึง 1000000f แล้วจะเริ่มทำต่อที่ Address 10000010 จนถึง 1000001f เป็นเช่นนี้ไปเรื่อยๆ จนถึง Address ที่ 1000ffff แต่เมื่อเรา โหลดโปรแกรม sample1.asm เข้าไปที่ QtSpim โดยที่ยังไม่ได้ Run นั้นจะมีการกำหนดค่าไปที่ Address เป็น Array of int ที่บรรจุค่า int 4 จำนวน นั้นคือ 10010001 เป็น 100, 10010002 เป็น -200, 10010003 เป็น 500, 10010004 เป็น 0 และ Address ตัวต่อไปจนถึง 1003ffff มีค่าเป็น 0

นางสาวจิรัชฌา วิทยาชีวะ 6010502519  
นางสาวจรรวี ทองเจริญ 6010504660

## Int Regs :

อธิบาย : ไม่มีการเปลี่ยนแปลงค่าใน  
Register เนื่องจากยังไม่ได้ Run

Int Regs [10]	
PC	= 0
EPC	= 0
Cause	= 0
BadVAddr	= 0
Status	= 805371664
HI	= 0
LO	= 0
R0 [r0]	= 0
R1 [at]	= 0
R2 [v0]	= 0
R3 [v1]	= 0
R4 [a0]	= 1
R5 [a1]	= 2147483148
R6 [a2]	= 2147483156
R7 [a3]	= 0
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
R26 [k0]	= 0
R27 [k1]	= 0
R28 [gp]	= 268468224
R29 [sp]	= 2147483144
R30 [s8]	= 0
R31 [ra]	= 0

## 1.3

## Text :

```

                                User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29)          ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4       ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4       ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2         ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2       ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop                  ; 189: nop
[0040001c] 3402000a ori $2, $0, 10        ; 191: li $v0 10
[00400020] 0000000c syscall              ; 192: syscall # syscall 10 (exit)
[00400024] 3c081001 lui $8, 4097 [my_array] ; 10: la $8, my_array # move address of my_array (&my_array) into $8
[00400028] 8d090000 lw $9, 0($8)          ; 11: lw $9, 0($8) # load my_array[0] into $9
[0040002c] 8d0a0004 lw $10, 4($8)         ; 12: lw $10, 4($8) # load my_array[1] into $10
[00400030] 012a5820 add $11, $9, $10      ; 13: add $11, $9, $10 # add the two numbers into $11
[00400034] 8d090008 lw $9, 8($8)          ; 14: lw $9, 8($8) # load my_array[2] into $9
[00400038] 01695820 add $11, $11, $9      ; 15: add $11, $11, $9 # add the number in $11 to it and accumulate to $11
[0040003c] ad0b000c sw $11, 12($8)        ; 16: sw $11, 12($t0) # store the added result to my_array[3]
[00400040] 3402000a ori $2, $0, 10        ; 20: li $v0, 10
[00400044] 0000000c syscall              ; 21: syscall

```

อธิบาย : จากภาพเป็นโค้ดในส่วนของ text เมื่อโหลดโปรแกรม sample1.asm แล้วกดรัน จะเห็นว่าโค้ดที่ปรากฏในส่วนของ text ไม่มีการเปลี่ยนแปลง

## Data :

```

User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 0000000100 -200 0000000500 0000000000 d . . . 8 . . . . .
[10010010]..[1003ffff] 00000000

```

อธิบาย : จากภาพเป็นการจองพื้นที่ใน Memory โดยถูกกำหนดให้ค่าเริ่มต้น(initial) ใน Memory นั้นมีค่าเป็น 00000000 โดยจากภาพ Address ใน Memory ที่เริ่มต้นที่ 10000000 ถึง 1000000f แล้วจะเริ่มทำต่อที่ Address 10000010 จนถึง 1000001f เป็นเช่นนี้ไปเรื่อยๆ จนถึง Address ที่ 1000ffff แต่เมื่อเรา โหลดโปรแกรม sample1.asm เข้าไปที่ QtSpim โดยที่ยังไม่ได้ Run นั้นจะมีการกำหนดค่าไปที่ Address เป็น Array of int ที่บรรจุค่า int 4 จำนวน นั้นคือ 10010001 เป็น 100, 10010002 เป็น -200, 10010003 เป็น 500, 10010004 เป็น 0 และ Address ตัวต่อไปจนถึง 1003ffff มีค่าเป็น 0

**Int Regs :**

อธิบาย : มีการนำค่าจาก my\_array มาใช้  
โดยเปลี่ยนแปลงตาม code ในไฟล์

sample1.asm โดยค่าที่เปลี่ยนแปลงมีดังนี้

- ค่า R8 ได้จาก Address ของ my\_array
- R9 เกิดจากการโหลดค่าจาก my\_array  
ตัวที่ 1
- R10 เกิดจากการโหลดค่าจาก my\_array  
ตัวที่ 2
- R11 เกิดจากการบวกค่าของ R9 และ  
R10 หลังจากนั้น ได้มีการเปลี่ยนแปลงค่า  
R9 โดย โหลดค่า R9 จาก my\_array  
ตัวที่ 3 และนำมาบวกกับ R11 อีกครั้ง  
ทำให้ R11 มีค่า 400 ดังภาพ

Int Regs [10]	
PC	= 4194372
EPC	= 0
Cause	= 0
BadVAddr	= 0
Status	= 805371664
HI	= 0
LO	= 0
R0 [r0]	= 0
R1 [at]	= 0
R2 [v0]	= 10
R3 [v1]	= 0
R4 [a0]	= 1
R5 [a1]	= 2147483156
R6 [a2]	= 2147483164
R7 [a3]	= 0
R8 [t0]	= 268500992
R9 [t1]	= 500
R10 [t2]	= -200
R11 [t3]	= 400
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
R26 [k0]	= 0
R27 [k1]	= 0
R28 [gp]	= 268468224
R29 [sp]	= 2147483152
R30 [s8]	= 0
R31 [ra]	= 4194328

## 1.4

### Hex

```
Int Regs [16]
PC      = 400044
EPC     = 0
Cause   = 0
BadVAddr = 0
Status  = 3000ff10

HI      = 0
LO      = 0

R0 [r0] = 0
R1 [at] = 0
R2 [v0] = a
R3 [v1] = 0
R4 [a0] = 1
R5 [a1] = 7fffffe14
R6 [a2] = 7fffffe1c
R7 [a3] = 0
R8 [t0] = 10010000
R9 [t1] = 1f4
R10 [t2] = ffffffff38
R11 [t3] = 190
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
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 10008000
R29 [sp] = 7fffffe10
R30 [s8] = 0
R31 [ra] = 400018
```

```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 00000064 ffffffff38 000001f4 00000190 d...8.....
[10010010]..[1003ffff] 00000000
```

### Dec

```
Int Regs [10]
PC      = 4194372
EPC     = 0
Cause   = 0
BadVAddr = 0
Status  = 805371664

HI      = 0
LO      = 0

R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 10
R3 [v1] = 0
R4 [a0] = 1
R5 [a1] = 2147483156
R6 [a2] = 2147483164
R7 [a3] = 0
R8 [t0] = 268500992
R9 [t1] = 500
R10 [t2] = -200
R11 [t3] = 400
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
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 268468224
R29 [sp] = 2147483152
R30 [s8] = 0
R31 [ra] = 4194328
```

```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 00000000100 -200 0000000500 0000000400 d...8.....
[10010010]..[1003ffff] 00000000
```

อธิบาย : จากภาพฐาน Decimal R10 มีค่าเท่ากับ -200 เมื่อเปลี่ยนเป็นฐาน Hexadecimal นั้น ต้องเปลี่ยนเป็น 2's complement แล้วบวก 1 เนื่องจากค่าในฐาน Decimal เป็นจำนวนเต็มลบ จึงทำให้ค่าในฐาน Hexadecimal คู่มีค่ามากกว่าฐาน Decimal