

```
#####
#          cs315 Week 12 - part (A)
#
#  ->  Hand assembly, branch address calculation
#
#####
Practice exercise:
    Given: first instruction "lw $t0, 0($sp)" is at address: "0x0040 0118"
```

```
* original code:
print_array:
    lw $t0, 0($sp)
    lw $t1, 4($sp)

print_array_loop:
    blez $t1, print_array_end

    lw $a0, 0($t0)
    li $v0, 1
    syscall

    addiu $t0, $t0, 4
    addiu $t1, $t1, -1

    b print_array_loop

print_array_end:
    jr $ra
```

```
step (1): expand macro instructions:
print_array:
    lw $t0, 0($sp)
    lw $t1, 4($sp)

print_array_loop:
    blez $t1, print_array_end

    lw $a0, 0($t0)
    ori $v0, $0, 1
    syscall

    addiu $t0, $t0, 4
    addiu $t1, $t1, -1

    bgez $0, print_array_loop

print_array_end:
    jr $ra
```

```
step (2): convert register names to register numbers:
print_array:
    lw $8, 0($29)
    lw $9, 4($29)

print_array_loop:
    blez $9, print_array_end

    lw $4, 0($8)
    ori $2, $0, 1
    syscall

    addiu $8, $8, 4
    addiu $9, $9, -1

    bgez $0, print_array_loop

print_array_end:
    jr $31
```

```
step (3): align the labels with assembly code:
print_array: lw $8, 0($29)
             lw $9, 4($29)

print_array_loop: blez $9, print_array_end

             lw $4, 0($8)
             ori $2, $0, 1
             syscall

             addiu $8, $8, 4
             addiu $9, $9, -1

             bgez $0, print_array_loop

print_array_end: jr $31
```

```
step (4): convert labels to addresses:
    print_array: lw $8, 0($29)          <-- 0x0040 0118
                  lw $9, 4($29)         <-- 0x0040 011C

    print_array_loop: blez $9, print_array_end <-- 0x0040 0120

                  lw $4, 0($8)          <-- 0x0040 0124
                  ori $2, $0, 1         <-- 0x0040 0128
                  syscall               <-- 0x0040 012C

                  addiu $8, $8, 4       <-- 0x0040 0130
                  addiu $9, $9, -1      <-- 0x0040 0134

                  bgez $0, print_array_loop <-- 0x0040 0138

    print_array_end: jr $31             <-- 0x0040 013C
```

```
step (5): calculate branch offsets:
    print_array: lw $8, 0($29)          <-- 0x0040 0118
                  lw $9, 4($29)         <-- 0x0040 011C
#
#    blez $9, print_array_end
#        distance = +7 (forward)
#        offset = distance - 1 = 6
#
    print_array_loop: blez $9, print_array_end <-- 0x0040 0120

                  lw $4, 0($8)          <-- 0x0040 0124
                  ori $2, $0, 1         <-- 0x0040 0128
                  syscall               <-- 0x0040 012C

                  addiu $8, $8, 4       <-- 0x0040 0130
                  addiu $9, $9, -1      <-- 0x0040 0134

#
#    bgez $0, print_array_loop
#        distance = -6 (backward)
#        offset = distance - 1 = -7
#
                  bgez $0, print_array_loop <-- 0x0040 0138

    print_array_end: jr $31             <-- 0x0040 013C
```

```
step (6): convert instructions to machine code:
# - address -           - machine code -

# 0x0040 0118      1000 11ss ssst tttt iiii iiii iiii iiii
# 0x0040 0118      1000 1111 1010 1000 0000 0000 0000 0000
    print_array: lw $8, 0($29)          <-- 0x0040 0118

# 0x0040 011C      1000 11ss ssst tttt iiii iiii iiii iiii
# 0x0040 011C      1000 1111 1010 1001 0000 0000 0000 0100
                  lw $9, 4($29)         <-- 0x0040 011C

#
#    blez $9, print_array_end
#        distance = 7 (forward)
#        offset = distance - 1 = 6
#
# 0x0040 0120      0001 10ss sss0 0000 iiii iiii iiii iiii
# 0x0040 0120      0001 1001 0010 0000 0000 0000 0000 0110
    print_array_loop: blez $9, print_array_end <-- 0x0040 0120

# 0x0040 0124      1000 11ss ssst tttt iiii iiii iiii iiii
# 0x0040 0124      1000 1101 0000 0100 0000 0000 0000 0000
                  lw $4, 0($8)          <-- 0x0040 0124

# 0x0040 0128      0011 01ss ssst tttt iiii iiii iiii iiii
# 0x0040 0128      0011 0100 0000 0010 0000 0000 0000 0001
                  ori $2, $0, 1         <-- 0x0040 0128

# 0x0040 012C      0000 00-- ---- ---- ---- ---- --00 1100
# 0x0040 012C      0000 0000 0000 0000 0000 0000 0000 1100
                  syscall               <-- 0x0040 012C

# 0x0040 0130      0010 01ss ssst tttt iiii iiii iiii iiii
# 0x0040 0130      0010 0101 0000 1000 0000 0000 0000 0100
                  addiu $8, $8, 4       <-- 0x0040 0130

# 0x0040 0134      0010 01ss ssst tttt iiii iiii iiii iiii
# 0x0040 0134      0010 0101 0010 1001 1111 1111 1111 1111
                  addiu $9, $9, -1      <-- 0x0040 0134

#
#    bgez $0, print_array_loop
#        distance = -6 (backward)
#        offset = distance - 1 = -7
#
# 0x0040 0138      0000 01ss sss0 0001 iiii iiii iiii iiii
```

```
# 0x0040 0138      0000 0100 0000 0001 1111 1111 1111 1001
    bgez $0, print_array_loop          <-- 0x0040 0138

# 0x0040 013C      0000 00ss sss0 0000 0000 0000 0000 1000
# 0x0040 013C      0000 0011 1110 0000 0000 0000 0000 1000
print_array_end: jr $31                <-- 0x0040 013C
```

--> Summary table:

#	- address -	- machine code (binary) -	- machine code (hex) -
#	0x0040 0118	1000 1111 1010 1000 0000 0000 0000 0000	0x8FA8 0000
#	0x0040 011C	1000 1111 1010 1001 0000 0000 0000 0100	0x8FA9 0004
#	0x0040 0120	0001 1001 0010 0000 0000 0000 0000 0110	0x1920 0006
#	0x0040 0124	1000 1101 0000 0100 0000 0000 0000 0000	0x8D04 0000
#	0x0040 0128	0011 0100 0000 0010 0000 0000 0000 0001	0x3402 0001
#	0x0040 012C	0000 0000 0000 0000 0000 0000 0000 1100	0x0000 000C
#	0x0040 0130	0010 0101 0000 1000 0000 0000 0000 0100	0x2508 0004
#	0x0040 0134	0010 0101 0010 1001 1111 1111 1111 1111	0x2529 FFFF
#	0x0040 0138	0000 0100 0000 0001 1111 1111 1111 1001	0x0401 FFF9
#	0x0040 013C	0000 0011 1110 0000 0000 0000 0000 1000	0x03E0 0008