Computer Architecture – LAB 8

kwp@hallym.ac.kr

OSLAB
Hallym University

LAB 8 - 1

- Reading and Copying a string..
  - Fill in the RED box

```
.data
ask:        .asciiz "Type a character string (<50 chars long): "
bye:        .asciiz "Press enter to exit..."
len:        .word 50
inputStr:   .space 50
copyStr:    .space 50

.text
.globl main
main:
    li $v0, 4           # system code 4: print a string
    la $a0, ask         # '$a0' will point out 'ask'
    syscall

    la $a0, inputStr    # '$a0' will point ouot 'inputStr'
    lw $a1, len         # Load word-0 from 'len' into '$a1'
    li $v0, 8           # system code 8: read a string
    syscall

    [                ]  # '$t2' will store the value 97 (97 == ASCII 'a')
                        # '$t3' will store the value 122(122 == ASCII 'z')
                        # '$t0' will store the value 0   ('$t0' is the index)
```

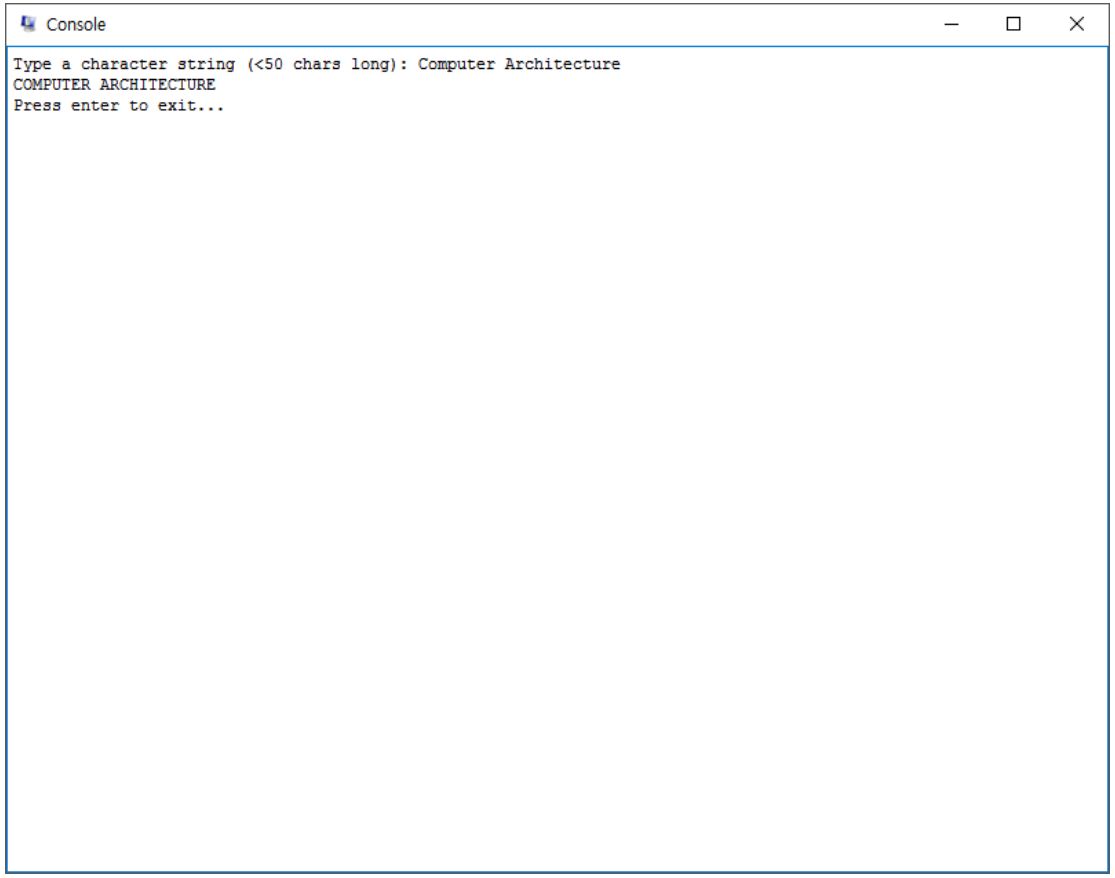
LAB 8 - 1

- Reading and Copying a string..
  - Fill in the RED box

```
loop:
    lb $t1, inputStr($t0)    # Load byte-$t0 from 'inputStr' into '$t1'
                             # Fetch a character from 'inputStre' into '$t1'
                             # if ($t1 < $t2) then go to the label 'skip'
                             # if ($t1 > $t3) then go to the label 'skip'

    addi $t1, $t1, -32       # $t1 <- $t1 - 32
                             # Conver to upper case
                             # The gap between lower case and upper case is 32
skip:
                             # Store a byte in '$t1' into byte-$t0 from 'copyStr'
                             # $t0 <- $t0 + 1 (Increase index)
                             # if ($t1 != 0) then go to the label 'loop'
finish:
    li $v0, 4        # system code 4: print a string
    la $a0, copyStr  # '$a0' will point out 'copyStr'
    syscall

    li $v0, 4        # system code 4: print a string
    la $a0, bye      # '$a0' will point out 'bye'
    syscall

    li $v0, 10       # exit program
    syscall
```

LAB 8 – 1

- Reading and Copying a string..
  - Result



LAB 8 – 2

- Convert Integer to Binary..
  - Fill in the RED box

```
.data
result: .space 32
        .byte 0
ask:    .asciiz "Enter decimal number: "
ansBin: .asciiz "Binary is: "
ansStr: .asciiz "\nBinary string: "
space:  .asciiz " "

.text
.globl main
main:
    li $v0, 4       # system code 4: print a string
    la $a0, ask     # '$a0' will point out 'ask'
    syscall

    li $v0, 5       # system code 5: read an integer value
    syscall         # READ
    move $t2, $v0   # $t2 <- $v0

    li $v0, 4       # system code 4: print a string
    la $a0, ansBin  # '$a0' will point out 'ansBin'
    syscall

    [          ]    # '$t4' will store the value 4 for a binary number
    [          ]    # We will device 32-digit into 8 groups consist of 4-digit
    [          ]    # '$t0' will store the value 32 for a binary string
                    # Because we will print a binary number which consist of 32-digit
                    # '$t3' will point out 'result'
```

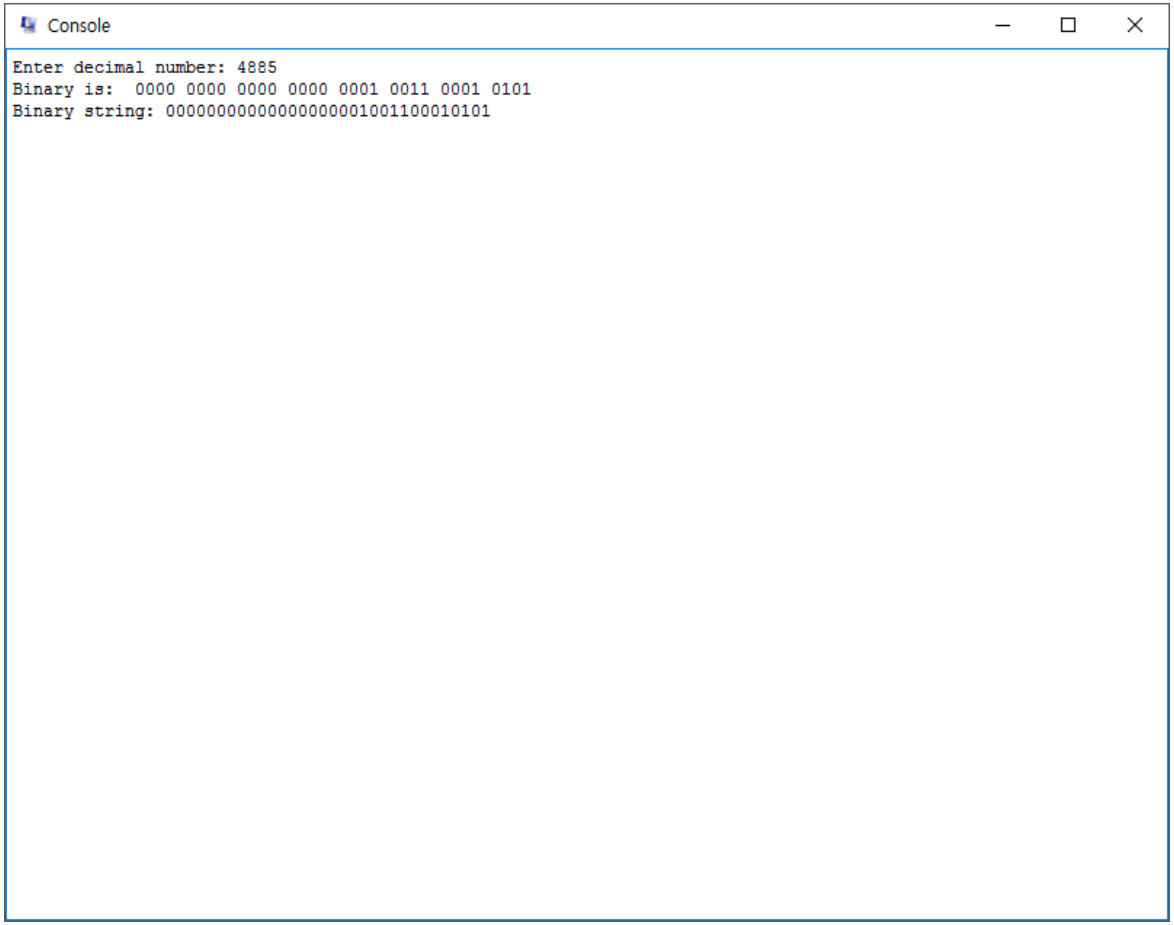
LAB 8 – 2

- Convert Integer to Binary..
  - Fill in the RED box

```
loop:
    rol $t2, $t2, 1      # $t2 <- $t2 << 1
                         # ROL(: Rotate left) is different with 'Shift instructions'
                         # After ROL, the leftmost number will be located at the rightmost place
                         # Start from the leftmost number
    andi $t1, $t2, 1     # $t1 <- $t2 AND 1

    # Print blank after 4-digit
    div $t0, $t4             # $t0 / $t4
    mfhi $t5                 # The remainder will move to '$t5' register
    bnez $t5, print_binary   # if ($t5 != 0) then go to the label 'print_binary'
    li $v0, 4                # system code 4: print a string
    la $a0, space            # '$a0' will point out 'space'
    syscall
print_binary:
    li $v0, 1            # system code 1: print an integer value
                         # $a0 <- $t1 for a binary number
                         # We will directly print a binary digit
    syscall

    # Store a digit into a string
                         # $t1 <- $t1 + 48  (The ASCII code of '0')
                         # Store a byte in '$t1' into byte-0 from '$t3'

    addi $t3, $t3, 1     # $t3 <- $t3 + 1   (Increase the index of 'result')
    addi $t0, $t0, -1    # $t0 <- $t0 - 1   (Decrease the count)
    bnez $t0, loop       # if ($t0 != 0) then go to the label 'loop'

    li $v0, 4            # system code 4: print a string
    la $a0, ansStr       # '$a0' will point out 'ansStr'
    syscall

    li $v0, 4            # system code 4: print a string
    la $a0, result       # '$a0' will point out 'result'
    syscall

    li $v0, 10           # system code 10: exit program
    syscall
```

LAB 8 – 2

- Convert Integer to Binary..
  - Result



TASK

- 과제
  - LAB 8-1, 8-2 를 완성하여 워드문서에 정리하여 제출
- 파일명
  - ca_08_학번_이름.docx
- 제출기한
  - 11월 30일 23:59까지
- **수업시간 내 완료시 조교의 확인을 받고 퇴실가능, 미확인시 결석처리**