MIPS Programming Note

Program Layout

- Data segment (.data)
  - Data label (msg_str:) +
  - data type (.asciiz) +
  - Data (“Enter some…”)
- Text segment (. text)
  - .globl main
  - Main label (main:)
  - Instruction label-1 (fac:)
  - Instruction label-2 (loophead:)

```
.data
msg_str:            .asciiz "Enter some number: "
```

```
.text

.globl main

main:
    la      $a0, msg_str            # load the ad
    li      $v0, 4                  # prepare for
    syscall                         # syscall 4
    li      $v0, 5                  # prepare for
 .....
 .....
fac:
    bne     $a0, $zero, loophead# If input is
    li      $v0, 1
    jr      $ra                     # return valu

loophead:
    move    $v0, $a0                # copy input
        until loop ends
    li      $t0, 1                  # load value
```

Data type

- **.word (int)**
- .half (short)
- .byte (char)
- **.asciiz (null terminated)**
- .ascii
- .double
- .float

Registers

- $zero
- $v0, $v1
- $a0 ~ $a3
- $t0 ~ $t9
- $s0 ~ $s7
- $sp, $fp
- $ra

Syscall (kernel mode, OS service)

print integer	**1**	**$a0 = integer to print**	
print float	2	$f12 = float to print	
print double	3	$f12 = double to print	
print string	4	$a0 = address of null-terminated string to print	
read integer	**5**		**$v0 contains integer read**
read float	6		$f0 contains float read
read double	7		$f0 contains double read
read string	8	$a0 = address of input buffer $a1 = maximum number of characters to read	
sbrk (allocate heap memory)	**9**	**$a0 = number of bytes to allocate**	**$v0 contains address of allocated memory**
exit (terminate execution)	**10**		

Find N! with function call & program control

```
main:
    la      $a0, msg_str
    li      $v0, 4
    syscall
    li      $v0, 5
    syscall
    move    $a0, $v0
        function call
    jal     fac
    move    $a0,$v0
        syscall 1
    li      $v0,1
    syscall
    li      $v0,10
    syscall
```

```
fac:
    beq     $a0,$zero,ret_one

    addi    $sp,$sp,-8
        call
    sw      $a0,0($sp)
    sw      $ra,4($sp)
    addi    $a0,$a0,-1
    jal     fac
    lw      $t0,0($sp)
    lw      $ra,4($sp)
    addi    $sp,$sp,8

    mul     $v0,$v0,$t0
    j       ret
ret_one:
    li      $v0,1
ret:
    jr      $ra
```

Find $Q1^3$ with memory allocation

```
.data
Q1:         .word 1, 1, 1, 0
msg_str: .asciiz "Q3 =\n"
.text
.globl main
main:
la   $a0, Q1
la   $a1, Q1
jal  mmul
move      $a0, $v0
la   $a1, Q1
jal       mmul
move      $t0, $v0
```

Result are in memory:
0($t0), 4($t0), 8($t0), 12($t0)

```
mmul:
move      $t0, $a0
move      $t1, $a1
li   $a0, 16
li   $v0, 9
syscall
```

(Calculate 1st element…)

```
sw   $t7, 0($v0)
```

(Calculate 2nd element…)

```
sw   $t7, 4($v0)
```

(Calculate 4th element…)

```
sw   $t7, 12($v0)
```

(Calculate 3rd element…)

```
sw   $t7, 8($v0)
jr   $ra
```