

Một số điều cần chú ý trong MIPS

- Khi sử dụng trong chương trình con cần phải push các biến đó vào stack rồi cuối làm hàm pop. **Để tránh thay đổi để các biến toàn cục ở các hàm phía trên**

Menu:

```
addi    $sp, $sp, -12
sw      $t0, 0($sp)
sw      $ra, 4($sp)

jal      Input    # Input day, month, year
sw      $v0, 8($sp)

la $a0, p_time
lw $a1, 8($sp)
jal Strcpy
```

Đoạn code cho thấy được việc có thể các vùng của stack để lưu tạm các giá trị hàm khác qua câu lệnh : **sw \$v0, 8(\$sp)**

- Khi cần xuất một biến description và chuỗi ta thực hiện cú pháp sau:

```
la      $a0, result
li      $v0, 4
syscall

lw      $a0, ($sp)
li      $v0, 4
syscall
```

- Viết một hàm cần comment các tham chiếu và biến xuất ra

```
#-----
strcpy: #char* Strcpy(char* dest, char* source )
        #a0 = destination , a1 = source
        addi    $sp, $sp, -16
        sw      $ra, 12($sp)
        sw      $s0, 8($sp)
        sw      $s1, 4($sp)
        sw      $t0, 0($sp)

        la      $s0, ($a0)
        la      $s1, ($a1)
```

- Để copy từ một chuỗi sang một chuỗi khác ta dùng VD copy theo DD/MM/YY

Date: # char* Date(char* day, char* month, char* year, char* TIME)

a0 = day , a1 = month , a2 = year , a3 =dd/mm/yyyy

addi \$sp, \$sp, -20

sw \$ra, 16(\$sp)

sw \$a0, 12(\$sp)

sw \$a1, 8(\$sp)

sw \$a2, 4(\$sp)

sw \$a3, 0(\$sp)

move \$a0, \$a3

lw \$a1, 12(\$sp)

jal Strcpy

move \$a0, \$v0 # a[] = 'DD'

li \$t0, 47 # a[2] = 47 = '/'

sb \$t0, 2(\$a0) # a[] = 'DD/'

la \$a0, 3(\$a0) # Shift left << 3

lw \$a1, 8(\$sp) # v0[] = 'MM'

jal Strcpy

la \$a0, -3(\$v0) # a[] 'DD/MM'

- Trong trường hợp muốn cắt chuỗi s chỉ lấy một phần từ nửa khoảng [0, len) giống C thì ta cần thực cần lệnh sau vd **len = 12**

```
la    $a0, 0($zero)
```

```
li    $t0, 0 # \0
```

```
sb    $t0, 12($a0)
```