



Arquitecturas y Organización de Computadoras I
2° Cuatrimestre

TP N° 8 – Programación en lenguaje ensamblador MIPS
con MIPSX



Objetivo: Comprender la estructura de un programa en lenguaje ensamblador MIPS, convención de llamada a procedimientos y funciones.

Recursos y Bibliografía:

Arq. MIPS Vol I, II and III.

Programa mipsx desarrollado por la cátedra.

Apunte MIPS, sección 9.

- 1) Leer el apunte de MIPS, sección 6 y la presentación de convención de llamada a procedimientos y funciones, y la presentación: <http://se.fi.uncoma.edu.ar/ayodc1/apuntes/llamadaAFunciones.pdf>
- 2) ¿Qué fallos comete el siguiente código de función en cuanto a la convención de uso de registros y llamada a procedimiento y funciones?

fun:

```
addiu    $sp, $sp, -32
sw       $ra, 28($sp)
sw       $fp, 24($sp)
move     $fp, $sp
li       $s0, 1
li       $t1, 2
jal      fun
addi     $k0, $v0, 3
add      $v0, $t1, $k0
add      $v0, $s1, $k0
move     $sp, $fp
lw       $ra, 28($sp)
lw       $fp, 24($sp)
addiu    $sp, $sp, 32
jr       $ra
```

- 3) Implementar una función que multiplique dos valores pasados por parámetros, respetando las convenciones de uso de registros y llamadas a funciones y procedimientos.
- 4) Implemente un programa que llame a la función desarrollada en el punto 3, respetando las convenciones de uso de registros y llamadas a funciones y procedimientos.
- 5)
 - a) Traduzca al lenguaje ensamblador el programa en C que se encuentra debajo (respeta la convención de llamada a procedimientos). Considere que char ocupa un byte e int una palabra.
 - b) Verifique su correcto funcionamiento en mipsx.
 - c) Responda: ¿por qué las variables “x” y “n” de pow no son las mismas que las variables globales “x” y “n” definidas antes de main?. Indique cuales son las direcciones efectivas de “x” y “n” global, y de las variables “i” y “pow” en pow().

```
int pow(char x, char n)
{
    int i;
    int pow = 1;

    for (i=0; i<n; i++)
        pow = pow * x;

    return pow;
}
```

```
char x = 30;
char n = 55;
int resultado = 0;
```

```
void main(void)
{
    resultado = pow(x, n);
}
```