



Arquitecturas y Organización de Computadoras I 2° Cuatrimestre



Facultad de
Informática

TP N° 7 – Programación en lenguaje ensamblador MIPS con MIPSX

Objetivo: Comprender la estructura de un programa en lenguaje ensamblador MIPS junto con la convención de llamada a procedimientos y funciones. Comprender el funcionamiento general de una organización de memoria caché y su implicancia.

Recursos y Bibliografía:

Arq. MIPS Vol I, II and III.

Programa mipsx.

Apunte MIPS, sección 6.

Caché: Patterson y Hennessy. Computer Organization and Design. The hw/sw interface.

1. a. ¿Qué fallos comete el siguiente código de función (o subrutina)¹**[*]** 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
```

b. ¿Qué problemas pueden suceder si un compilador mal desarrollado (o un programador) no utiliza la convención de llamadas a procedimientos?. Mencione tres problemas que pueden surgir.

2. Traduzca el programa en C, que se encuentra debajo, a un programa en lenguaje ensamblador MIPS, como lo haría un compilador. Respete la convención de llamada a procedimientos. Considere que char ocupa un byte e int una palabra. Verifique su correcto funcionamiento en mipsx.

- ¿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()**.
- ¿Cuál es la dirección del último byte del segmento de texto? ¿Cuántas pseudo instrucciones tiene el programa?
- Indique las direcciones de inicio de los segmentos de pila, segmento de texto y

¹**[*]** Subrutina o función: [https://en.wikipedia.org/wiki/Function_\(computer_programming\)](https://en.wikipedia.org/wiki/Function_(computer_programming))

segmento de datos del programa.

```
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);
}
```

3. Desarrollar una función que reemplace las letras mayúsculas por minúsculas de una cadena ASCII (la cadena puede contener caracteres que no sean letras). La función recibe como parámetro de entrada un puntero (un entero que contiene la dirección de una variable) de la cadena a ser convertida. Realice el reemplazo sobre la cadena original. Respete la convención de llamada a procedimiento. Verifique el funcionamiento de su subrutina desarrollada.

4. Ejecutar los programas m1.c y m2.c, vistos en la clase de teoría, en las PCs de laboratorio. Suponiendo que en m1 la tasa de aciertos a caché se aproxima al 90%. Realice un análisis comparando los tiempos de ejecución, y estime cuál es el tiempo medio de acceso a memoria en la PC de laboratorio en la que está trabajando.

5. Se dispone de una caché asociativa por conjuntos de dos vías, con 8 entradas de caché. En cada línea de caché se almacena una palabra (en cada vía). Se realizó una acción de anular la caché, por lo que todos los bits de validez de las entradas están en cero. Se ejecuta el siguiente programa:

```
unsigned char  blue[20] = { 23, 3, 8, 9, 21, 0, 2, 9};
int  ahorro[6] = {0, 0, 0, 0, 0, 0};
unsigned char  inc[20] = { 5, 4, 3, 56, 56, 57, 58, 59};
unsigned char oficial = 5;
int i;
int res = 0;

main ()
{
    for (i=0; i<8; i++) {
        res = res + blue[i] + inc[i];
    }
    res = res + oficial;
```

```
    res = res + blue[0] + inc[0];  
}
```

- a. Ejecute en mipsx (traduzca como lo haría un compilador). Indique a qué líneas de caché corresponden los accesos a memoria. ¿Cuántos fallos y aciertos suceden?. ¿Cuál es el tiempo de acceso medio si el procesador es de 2Mhz, un acceso a caché toma de 2 microsegundos y un acceso a memoria toma 60 microsegundos?.
- b. ¿Qué beneficios podría tener una caché de este estilo sobre una caché de mapeo directo de 8 entradas y una palabra en cada línea de caché?. Presente un ejemplo de código en ensamblador donde se refleje la diferencia entre ambas (donde una sea preferible sobre la otra).