Organización y Arquitectura de Computadoras

6 Septiembre 2016

1 Lenguaje Ensamblador

En esta práctica vamos a conocer las partes básicas de un lenguaje ensamblador. Vamos a utilizar un simulador del lenguaje MIPS llamado SPIM. Este es un lenguaje RISC con puras operaciones Registro-Registro.

- Instrucción: Esta se conforma del nombre de la operación seguida por sus argumentos. En esta práctica revisaremos cinco operaciones: suma (add), resta (sub), cargar (lw), guardar (sw) y mover (move).
- Directiva: Es una instruccn especial dada al ensamblador, no es una instruccion como tal, todas las directivas de ensamblador (este es su nombre completo) empiezan con un punto. Las directivas que vamos a ver en esta practica son: .data, .word y .text.
- Etiquetas: Cualquier linea de nuestro programa puede ser precedida por una etiqueta seguida por dos puntos (:). Esta etiqueta sigue las mismas reglas que en la mayoria de los lenguajes de programacion: Se conforma por una letra seguida de una cadena de letras, dgitos y/o guiones bajos. Por ejemplo: etiqueta_1: .word 32.\

  Las unicas restricciones son para las etiquetas son: no se pueden repetir etiquetas dentro de uno o mas programas cargados en memoria y no se pueden usar palabras reservadas para instrucciones. El objetivo de las etiquetas es tener una referencia a alguna instruccin o linea de nuestro programa. Es decir representan la direccin de memoria en la cual esta la instruccion.
- Comentarios: Cualquier linea puede tener un comentario al nal. El cual esta precedido por un signo de gato #. MIPS ignora el resto de la linea despues de ver un #.
- Etiqueta main: Asi como en un programa escrito en Java o en C se tiene que escribir un procedimiento main(). En un programan MIPS deben de especicar una instruccin con etiqueta main. La cual va a indicar el inicio de su programa, es decir la primera instruccion a ejectutar.

- Terminación del programa (Salto de retorno): Al terminar el programa, SPIM espera que le regresemos el control. Y esto involucra hacer un salto a la direccion de retorno. Por tanto todos los programas que escriban en SPIM tienen que terminar con un salto a la direccion de retorno (que se guarda en $ra). Esto se hace con la siguiente isntruccion:

  jr $ra

- Registros Enteros:MIPS tiene 32 registros enteros de proposito general, asi como 64 dedicados a punto otante que revisaremos en otra practica. Salvo algunas excepcio- nes (vgr: 0 y 1); no existe ninguna restriccion en el uso de estos registros, de aqui el nombre de proposito general, aunque existen convenciones de uso (que por el bien de la comunidad deben respetar).

2 Ejemplo

SUMA

1. .data
2. w: . word 0 x42
3. x : . word 0 x43
4. y : . word 0 x44
5. z : . word 0 x45
6. #
7. .t e x t
8. main :
9. lw $s0 , w # s0< = w
10. lw $s1 , x # s1< = x
11. lw $s2 , y # s2< = y
12. lw $s3 , z # s3< = z
13. add $t0, $s0, $s1

3 Ejercicios

1. Implementa en un archivo como le harias si para copiar el valor de un registro a otro sin usar la instruccion move.

2. En un archivo escribe un programa que calcule

   $x = y + z$

   $x = (z + w) + (z + (y + (w - v)))$

   $x = (w - z) + (x + 23)$

   $x = v + 0xef1$

4 Fecha de entrega

Miercoles 21 de Septiembre.