EJERCICIOS DE PRACTICAS DE ENSAMBLADOR MIPS
ESTRUCTURA / ORGANIZACIÓN DE COMPUTADORES
BOLETÍN Nº 1

Ejercicio 1: (Operaciones básicas con la memoria)

Estudiar el siguiente programa en ensamblador del MIPS, que carga en la memoria los datos siguientes: los bytes correspondientes a la cadena "!Hola Mundo!" y a continuación los datos en forma de bytes: 5,10,15,20,25,30,35,40. Consultar las transparencias 8 y 16 del tutorial del lenguaje ensamblador del MIPS.

```
# Prácticas ensamblador MIPS
# Estructura de Computadores. 2° ITIS CUM-UEX 2004/2005
# Ejercicio 1
      .data
      .asciiz "¡Hola Mundo!\n"
      .byte 5,10,15,20,25,30,35,40
      .text
```

- *Cargar el programa en el simulador.*
- *Interpretar el contenido de las direcciones de memoria (consultar las transparencias 14, 15 y 26 del tutorial y también la tabla de código ASCII).*

La siguiente tabla puede servir de ayuda:

ASCII	!	H	o	l	a		M	u	n	d	o	!	\n	NULL
Decimal	33	72	111	108	97	32	77	117	110	100	111	33	10	0
Hexa-decimal	21	48	6f	6c	61	20	4d	75	6e	64	6f	21	0a	00

Ejercicio 2: (Operaciones básicas con la memoria)

Realizar un programa en ensamblador del MIPS que cargue en la memoria, consecutivamente, los siguientes datos:

- *Los siguientes bytes: 16, 32, 48.*
- *Las siguientes medias palabras: 16, 32, 48.*
- *Las siguientes palabras:16, 32, 48.*

Cargar el programa en el simulador e interpretar el contenido de las direcciones de memoria.

Ejercicio 3: (Arrays y constantes en memoria. Carga de memoria a registros)

Cargar en el simulador SPIM el siguiente programa:

```
# Prácticas ensamblador MIPS
# Estructura de Computadores. 2º ITIS CUM-UEX 2004/2005
# Ejercicio 3

        .data               # *** SEGMENTO DE DATOS ***
valor: .word 5,10,15,20     # Defino array 4 palabras (decimal).
                            # valor[2] es 15 = 0xf
ind:    .word 1             # Usado como índice del array "valor"
        .byte 0x1a,0x0b,10  # Defino los 3 primeros bytes de la
                            # siguiente palabra (hex. y dec.).
        .align 2            # Alineo el siguiente dato en memoria
                            # para que esté alineado en palabra
                            # (empiece en una palabra)
        .ascii "Hola"       # Cadena de caracteres
        .asciiz",MIPS"      # Cadena de caracteres terminada
                            # con el caracter nulo.

#-------------------------------------------------------------------------------
        .text               # *** SEGMENTO DE TEXTO ***
        .globl main         # Etiqueta main visible a otros ficheros
main:                       # Rutina principal
        lw $s0,valor($zero) # Carga en $s0 valor[0]. ($s0 <-- valor[0])
                            # También vale: lw $s0,valor

        lw $s4,ind          # $s4 <-- índice del array
        mul $s5,$s4,4       # $s5 <-- ind*4
        lw $s1,valor($s5)   # $s1 <-- valor[1]

        add $s4,$s4,1       # Incrementamos el índice del array
        mul $s5,$s4,4
        lw $s2,valor($s5)   # $s2 <-- valor[2]

        add $s4,$s4,1       # Incrementamos el índice del array
        mul $s5,$s4,4
        lw $s3,valor($s5)   # $s3 <-- valor[3]
```

- *Comprobar en la ventana de mensajes que la carga ha sido correcta.*
- *Comprobar en la ventana del segmento de datos que todos los bytes que aparecen en las palabras de memoria corresponden con las definiciones de datos que aparecen en las directivas del código ensamblador.*
- *Estudiar el contenido de la ventana de registros. Observar los registros que aparecen y sus valores (ver el listado de los registros del MIPS y sus distintos valores en la transparencia nº 9 del tutorial).*
- *Estudiar el contenido de la ventana del segmento de texto.*
- ***Escribir el valor que deberá tomar cada registro tras ejecutarse cada línea de código.***
- *Ejecutar el programa cargado en SPIM.*
- *Comprobar que los valores que van tomando los registros al ejecutarse las operaciones descritas en las líneas de código son los esperados.*
- *Estudiar cómo se cargan en registros las distintas posiciones de un array de datos en memoria y qué operaciones aritméticas es necesario realizar (y por qué).*
- *Estudiar las directivas e instrucciones que aparecen en el código (ver transparencia nº 8 del tutorial).*

Ejercicio 4: (Aplicación de arrays y ctes. en memoria. Carga de memoria a registros)

Realizar un programa en ensamblador del MIPS que cargue en un array de datos en memoria (etiquetado como "array") los siguientes enteros: 2, 4, 6, 8, 10. A continuación el programa debe cargar en el registro t0 el dato array[2] y en el registro t1 el dato array[4].

Ejercicio 5: (Modos de direccionamiento)

Realizar un programa en ensamblador del MIPS que cargue el contenido de una posición de memoria en un registro y luego escriba el valor del registro en consola. Debe repetir esta operación para cada uno de los 7 modos de direccionamiento existentes en el MIPS (estudiar la transparencia nº 10 del tutorial). Comprobar que en la consola aparece siempre el mismo valor (pues son modos distintos de acceder a un mismo dato). Se recomienda enseñar al profesor el código desarrollado, con el objeto de identificar correctamente cada modo de direccionamiento.