



Mireles Martine Angel Mauricio	INSTITUTO POLITÉCNICO NACIONAL	ESCUELA SUPERIOR DE CÓMPUTO
2	TAREA 9 SEGUNDO PARCIAL	22-04-2022
3CM13	ARQUITECTURA DE COMPUTADORAS	PERIODO ESCOLAR 2022-2

CONVERSIÓN MIPS A CÓDIGO MÁQUINA

2.- Escribir en código máquina los ejercicios de las tareas 3, 4, y 5, segundo parcial desarrolladas en QtSpim
Como ejemplo véase el caso de una instrucción particular

Expresión en texto	ori rt, rs, imm			
Expresión en código simbólico de un caso particular	0xd \$16, \$17, 0x478			
Formato del registro	opcode	rs	rt	imm
	0xd	R17	R16	0x478
Código binario separado en campos	001101 10001 10000 0000000111011110			
Código binario	00110110001100000000000111011110			
Código binario separado en campos	0011 0110 0011 0000 0000 0001 1101 1110			
Código hexadecimal	3 6 3 0 0 1 d E			

Expresión en texto	mul rd, rs, rt					
Expresión en código simbólico de un caso particular	0x1c \$04, \$04, 0x2					
Formato del registro	opcode	rs	rt	rd	0	2
	0x1c	R4	R4	0x2	0x0	0x2
Código binario separado en campos	011100 00100 00100 00010 00000 000010					
Código binario	01110000100001000001000000000010					
Código binario separado en campos	0111 0000 1000 0100 0001 0000 0000 0010					
Código hexadecimal	7 0 8 4 1 0 0 2					

Expresión en texto	addi rt, rs, imm			
Expresión en código simbólico de un caso particular	0x08 \$04, \$0, 0xAF			
Formato de registro	opcode	rs	rt	imm
	0x08	R0	R4	0xAF
Código binario separado en campos	001000 00000 00100 0000000010101111			
Código binario	00100000000001000000000010101111			
Código binario separado en campos	0010 0000 0000 0100 0000 0000 1010 1111			
Código hexadecimal	2 0 0 4 0 0 A F			



Expresión en texto	addi rt, rs, imm			
Expresión en código simbólico de un caso particular	0x08 \$2, \$0, 0x1			
Formato de registro	opcode	rs	rt	imm
	0x08	R0	R2	0x1
Código binario separado en campos	001000 00000 00010 0000000000000001			
Código binario	00100000000000100000000000000001			
Código binario separado en campos	0010 0000 0000 0010 0000 0000 0000 0001			
Código hexadecimal	2 0 0 2 0 0 0 1			

Expresión en texto	syscall			
Expresión en código simbólico de un caso particular	0x0			
Formato del registro	opcode	0x0	0x0	0xc
	0x0	0x0	0x0	0xc
Código binario separado en campos	000000 00000000000000000000 001100			
Código binario	0000000000000000000000000000001100			
Código binario separado en campos	0000 0000 0000 0000 0000 0000 0000 1100			
Código hexadecimal	0 0 0 0 0 0 0 C			

a).- Código máquina de la tarea 3

main:

.text

```
addi $a0,$0,175    #Valor a escribir en $a0
addi $v0,$0,1      #Función 1, print integer
syscall            #Escribe en consola $a0
```

```
addi $v0, $0, 10    #Funtion 10, exit
syscall             #Acaba el programa
```

001000000000010000000000010101111

00100000000000100000000000000001

000000000000000000000000000001100

001000000000001000000000000001010

000000000000000000000000000001100

addi \$a0,\$0,175

addi \$v0,\$0,1

syscall

addi \$v0, \$0, 10

syscall

b).- Código máquina de la tarea 4

main:



```
.text
addi $a0,$0,40      #Valor a escribir en $a0
addi $a0,$a0,10     #Valor a escribir en $a0
addi $v0,$0,1       #Función 1, print integer
syscall             #Escribe en consola $a0

addi $v0, $0, 10     #Funtion 10, exit
syscall              #Acaba el programa

001000000000010000000000000101000    addi $a0,$0,40
001000001000010000000000000001010    addi $a0,$a0,10
001000000000001000000000000000001    addi $v0,$0,1
0000000000000000000000000000001100    syscall
001000000000001000000000000001010    addi $v0, $0, 10
0000000000000000000000000000001100    syscall
```

c).- Código máquina de la tarea 5

main:

```
.text
addi $a0,$0,40      #Valor a escribir en $a0
mul $a0,$a0,2        #Valor a escribir en $a0
addi $v0,$0,1       #Función 1, print integer
syscall             #Escribe en consola $a0

addi $v0, $0, 10     #Funtion 10, exit
syscall              #Acaba el programa

001000000000010000000000000101000    addi $a0,$0,40
01110000100001000001000000000010    mul $a0,$a0,2
001000000000001000000000000000001    addi $v0,$0,1
0000000000000000000000000000001100    syscall
001000000000001000000000000001010    addi $v0, $0, 10
0000000000000000000000000000001100    syscall
```