

Pregunta 1  
Completa  
Puntuación: 1,00  
sobre 1,00  
Marcar a pregunta

Registros

PC=00400000 EPC=00000000 Cause=00000000  
Status=0000ff03 BadVaddr=00000000  
HI=00000000 LO=00000000

Registros generales

\$zero=00000000	\$t0=00000000	\$s0=00000000	\$t8=00000000
\$at=00000000	\$t1=00000000	\$s1=00000000	\$s9=00000000
\$v0=00000000	\$t2=00000000	\$s2=00000000	\$k0=00000000
\$v1=00000000	\$t3=00000000	\$s3=00000000	\$k1=00000000
\$a0=00000000	\$t4=00000000	\$s4=00000000	\$gp=00000000
\$a1=00000000	\$t5=00000000	\$s5=00000000	\$sp=70000000
\$a2=00000000	\$t6=00000000	\$s6=00000000	\$s8=00000000
\$a3=00000000	\$t7=00000000	\$s7=00000000	\$ra=00000000

Registros de punto flotante

\$f0=00000000	\$f8=00000000	\$f16=00000000	\$f24=00000000
\$f1=00000000	\$f9=00000000	\$f17=00000000	\$f25=00000000
\$f2=00000000	\$f10=00000000	\$f18=00000000	\$f26=00000000
\$f3=00000000	\$f11=00000000	\$f19=00000000	\$f27=00000000
\$f4=00000000	\$f12=00000000	\$f20=00000000	\$f28=00000000
\$f5=00000000	\$f13=00000000	\$f21=00000000	\$f29=00000000
\$f6=00000000	\$f14=00000000	\$f22=00000000	\$f30=00000000
\$f7=00000000	\$f15=00000000	\$f23=00000000	\$f31=00000000

Segmento de datos

MEMORIA

[0x10010000]	0x7292091a	0xf3057b87	0x21000000	0xffffffff
[0x10010010]	0x98765432	0xac56b478	0x657e3acd	0xffffffff
[0x10010020]	0x394b3231	0x6162e6e8	0x69626163	0x00000064
[0x10010030]	0x00000064	0x2000c564	0x63626160	0x00000064
[0x10010040]	0x000000ff	0x00000076	0xde00fe61	0x2365dda

PILA

[0x70000000]

Diagrama de arquitectura MIPS

Diagrama de flujo de ejecución

Segmento de texto

[0x00400000]	0x3c080000	lui \$t0, 0x1001
[0x00400004]	0x35080018	ori \$t0, \$t0, 0x0018
[0x00400008]	0x8c100000	lw \$s0, 0(\$t0)
[0x0040000c]	0x20110001	addi \$s1, \$zero, 1
[0x00400010]	0x02308820	add \$s1, \$s1, \$s0
[0x00400014]	0x3c080000	lui \$t0, 0x1001
[0x00400018]	0x3508002b	ori \$t0, \$t0, 0x002b
[0x0040001c]	0x81100003	lb \$s0, 3(\$t0)
[0x00400020]	0x3c040000	lui \$a0, 0x1001
[0x00400024]	0x34840026	ori \$a0, \$a0, 0x0026
[0x00400028]	0x20020004	addi \$v0, \$zero, 4
[0x0040002c]	0x0000000c	syscall
[0x00400030]	0x2002000a	addi \$v0, \$zero, 10
[0x00400034]	0x0000000c	syscall

Volver editor

Paso anterior

Paso siguiente

Ciclo anterior

Ciclo siguiente

Ejecutar

Breakpoint

CICLOS

En esta imagen se muestra la ejecución en el simulador de un código MIPS. ¿Qué valor contendrá el registro \$s0 al finalizar de ejecutarse la instrucción lb \$s0, 3(\$t0)?

Seleccione una:

☐ a. 0x00000065

☐ b. 0x63626165

☐ c. 0x00000061

☒ d. 0x00000062

Puntuación: 1,00  
sobre 1,00

Marcar a  
pregunta

PC=00400000 EPC=00000000 Cause=00000000  
Status=0000ff03 BadVaddr=00000000  
HI=00000000 LO=00000000

Registros generales			
\$zero=00000000	\$t0=00000000	\$s0=00000000	\$t8=00000000
\$at=00000000	\$t1=00000000	\$s1=00000000	\$s9=00000000
\$v0=00000000	\$t2=00000000	\$s2=00000000	\$k0=00000000
\$v1=00000000	\$t3=00000000	\$s3=00000000	\$k1=00000000
\$a0=00000000	\$t4=00000000	\$s4=00000000	\$gp=00000000
\$a1=00000000	\$t5=00000000	\$s5=00000000	\$sp=70000000
\$a2=00000000	\$t6=00000000	\$s6=00000000	\$s8=00000000
\$a3=00000000	\$t7=00000000	\$s7=00000000	\$ra=00000000

Registros de punto flotante			
\$f0=00000000	\$f8=00000000	\$f16=00000000	\$f24=00000000
\$f1=00000000	\$f9=00000000	\$f17=00000000	\$f25=00000000
\$f2=00000000	\$f10=00000000	\$f18=00000000	\$f26=00000000
\$f3=00000000	\$f11=00000000	\$f19=00000000	\$f27=00000000
\$f4=00000000	\$f12=00000000	\$f20=00000000	\$f28=00000000
\$f5=00000000	\$f13=00000000	\$f21=00000000	\$f29=00000000
\$f6=00000000	\$f14=00000000	\$f22=00000000	\$f30=00000000
\$f7=00000000	\$f15=00000000	\$f23=00000000	\$f31=00000000

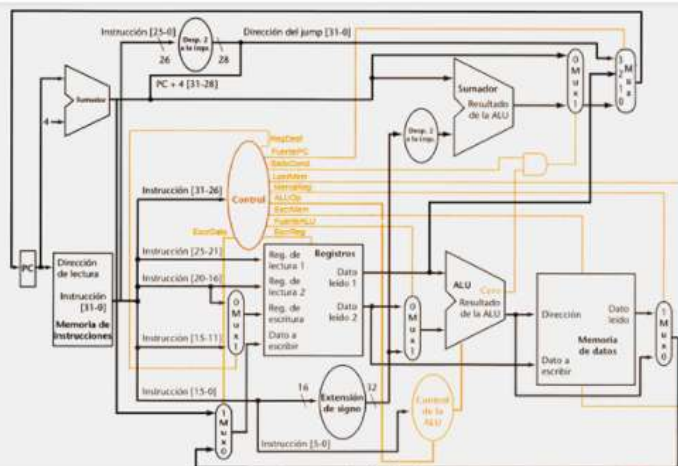
#### Segmento de datos

##### MEMORIA

[0x10010000]	0x7232091a	0xf3057b87	0x21000000	0xfffffff
[0x10010010]	0x98765432	0xac56b478	0x657e3acd	0xfffff
[0x10010020]	0x334b3231	0x61626a68	0x65006162	0x63626165
[0x10010030]	0x00000064	0x20006564	0x63626160	0x00000064
[0x10010040]	0x000000ff	0x00000076	0xde00fe51	0x2366ddaa


##### PILA


[0x70000000]




#### Segmento de texto


[0x00400000]	0x3c080000	lui \$t0, 0x1001
[0x00400004]	0x35080018	ori \$t0, \$t0, 0x0018
[0x00400008]	0x8d100000	lw \$s0, 0(\$t0)
[0x0040000c]	0x20110001	addi \$s1, \$zero, 1
[0x00400010]	0x02308820	add \$s1, \$s1, \$s0
[0x00400014]	0x3c080000	lui \$t0, 0x1001
[0x00400018]	0x3508002b	ori \$t0, \$t0, 0x002b
[0x0040001c]	0x81100003	lb \$s0, 3(\$t0)
[0x00400020]	0x3c040000	lui \$a0, 0x1001
[0x00400024]	0x34840026	ori \$a0, \$a0, 0x0026
[0x00400028]	0x20020004	addi \$v0, \$zero, 4
[0x0040002c]	0x0000000c	syscall
[0x00400030]	0x2002000a	addi \$v0, \$zero, 10
[0x00400034]	0x0000000c	syscall

 Volver editor


 Paso anterior

 Paso siguiente

 Ciclo anterior

 Ciclo siguiente

 Ejecutar

 Breakpoint

CICLOS

En esta imagen se muestra la ejecución en el simulador de un código MIPS. ¿Qué valor contendrá el registro \$s0 al finalizar de ejecutarse la instrucción lb \$s0, 3(\$t0)?

Seleccione unha:

- ☐ a. 0x00000065
- ☐ b. 0x63626165
- ☐ c. 0x00000061
- ☒ d. 0x00000062

A resposta correcta é:  
0x00000062

Pregunta 2

Completa

Puntuación: 1,00 sobre 1,00

⚑ Marcar a pregunta

Registers

PC=00400000    EPC=00000000    Cause=00000000  
Status=0000ff03    BadVaddr=00000000  
HI=00000000    LO=00000000

General purpose registers

\$zero=00000000	\$t0=00000000	\$s0=00000000	\$t8=00000000
\$at=00000000	\$t1=00000000	\$s1=00000000	\$s9=00000000
\$v0=00000000	\$t2=00000000	\$s2=00000000	\$a0=00000000
\$v1=00000000	\$t3=00000000	\$s3=00000000	\$a1=00000000
\$a0=00000000	\$t4=00000000	\$s4=00000000	\$gp=00000000
\$a1=00000000	\$t5=00000000	\$s5=00000000	\$sp=70000000
\$a2=00000000	\$t6=00000000	\$s6=00000000	\$s8=00000000
\$a3=00000000	\$t7=00000000	\$s7=00000000	\$ra=00000000

Floating point registers

\$f0=00000000	\$f8=00000000	\$f16=00000000	\$f24=00000000
\$f1=00000000	\$f9=00000000	\$f17=00000000	\$f25=00000000
\$f2=00000000	\$f10=00000000	\$f18=00000000	\$f26=00000000
\$f3=00000000	\$f11=00000000	\$f19=00000000	\$f27=00000000
\$f4=00000000	\$f12=00000000	\$f20=00000000	\$f28=00000000
\$f5=00000000	\$f13=00000000	\$f21=00000000	\$f29=00000000
\$f6=00000000	\$f14=00000000	\$f22=00000000	\$f30=00000000
\$f7=00000000	\$f15=00000000	\$f23=00000000	\$f31=00000000

Data segment

MEMORY

[0x10010000]	0x00000000	0x00000005	0x00000005
[0x10010010]	0x00000005	0x00000003	0x00000005
[0x10010020]	0x00000001		

STACK

[0x70000000]			
--------------	--	--	--

Text segment

[0x00400044]	0x01354820	add \$t1, \$t1, \$s5
[0x00400048]	0x8d280000	lw \$t0, 0(\$t1)
[0x0040004c]	0x1511000d	bne \$t0, \$s1, Fin
[0x00400050]	0x22100001	addi \$s0, \$s0, 1
[0x00400054]	0x08100015	j Bucle
[0x00400058]	0x00804020	add \$t0, \$a0, \$0
[0x0040005c]	0x00a04820	add \$t1, \$a1, \$0
[0x00400060]	0x00005020	add \$t2, \$0, \$0
[0x00400064]	0x11200005	beq \$t1, \$0, salir
[0x00400068]	0x8d0b0000	lw \$t3, 0(\$t0)
[0x0040006c]	0x016a5020	add \$t2, \$t3, \$t2
[0x00400070]	0x01306555	addi \$s1, \$s1, 1

Go back editor

Previous step

Next step

Previous cycle

Next cycle

Execute

Breakpoint

0x00400000

CYCLES

A partir de la codificación binaria de la instrucción **j Bucle**, indica cuál es la dirección destino del salto.

- ☐ a. 0x00400040
- ☐ b. 0x00400020
- ☒ c. 0x00400054
- ☐ d. 0x00400080

A resposta correcta é:  
0x00400054

Pregunta 3  
Completa  
Puntuación:  
-0,33 sobre 1,00  
⚑ Marcar a  
pregunta

En el código de la figura se ha saltado a una subrutina mediante la instrucción jal salto. Teniendo en cuenta que jal ha saltado a la instrucción cuya codificación es 0x3C081001 en la figura y que el código de operación de la instrucción jal es 3 en decimal:

- A resposta correcta é: El campo etiqueta de salto de la instrucción jal es de 26 bits y es 0x040002 y al final dos dígitos binarios 11.

Pregunta 4  
Completada  
Puntuación: 1,00  
sobre 1,00  
🚩 Marcar a pregunta

Suponiendo que el PC+4 (la dirección de memoria de la siguiente instrucción a ejecutar si no se produce un salto) es igual a 0x40000000, ¿Podemos saltar a la dirección 0x4001ffff con una instrucción **beq**?

- ☐ a. No, no es posible con una instrucción **beq** pero si con una **j**
- ☒ b. Sí, el campo salto de dicha instrucción es 01 1111 1111 1111 11
- ☐ c. Sí, debemos sumarle al PC+4 un total de 32.768 palabras
- ☐ d. No, no es posible con una instrucción **beq**

A respuesta correcta es:

Sí, el campo salto de dicha instrucción es 01 1111 1111 1111 11

Pregunta 5  
No respondida  
Puntuación como 1,00  
🚩 Marcar a pregunta

```
.text
.globl main
main:
    la $a0,array
    la $a1,count
    lv $a1, 0($a1)

    jal suma

    addi $a0, $v0, 0
    addi $v0, $0, 1
    syscall

    addi $v0,$0,10
    syscall

suma:
    add $t0, $a0, $0
    add $t1, $a1, $0
    add $t2, $0, $0
```

Dada la porción de código que se muestra en la figura y siguiendo al convenio que rige el proceso de guardar el contenido de registros en la pila cuando se llama a una subrutina:

- ☐ a. El programa no es correcto porque debería haber guardado en el programa principal el contenido de los registros \$t0, \$t1 y \$t2 en la pila antes de llamar a la subrutina.
- ☐ b. El programa es correcto ya que guardar el contenido de los registros tipo t en la pila, en caso de ser necesario, no es tarea del programa principal sino de la subrutina.
- ☐ c. El programa es correcto ya que la subrutina solo tiene que guardar en la pila, en caso de ser necesario, el contenido de los registros salvados, no el de los temporales.
- ☐ d. El programa no es correcto ya que la subrutina debería haber comenzado guardando el contenido de los registros \$t0, \$t1 y \$t2 en la pila.

A respuesta correcta es: El programa es correcto ya que la subrutina solo tiene que guardar en la pila, en caso de ser necesario, el contenido de los registros salvados, no el de los temporales.



Pregunta **6**

Completa

Puntuación: 1,00  
sobre 1,00

🚩 Marcar a  
pregunta

Dado el siguiente código ensamblador:

```
n2: .word 0x1111abcd
```

```
la $t2, n2
```

sabiendo que la palabra n2 está almacenada en la dirección de memoria:  
0x10010018

indicar a qué dos instrucciones MIPS equivaldría la pseudoinstrucción la.

Seleccione unha:

- ☒ a. lui \$t2, 0x1001  
ori \$t2, \$t2, 0x0018
- ☐ b. ori \$t2, \$t2, 0x1111  
lui \$t2, 0xabcd
- ☐ c. ori \$t2, \$t2 0x1001  
lui \$t2, 0x0018
- ☐ d. lui \$t2, 0x1111  
ori \$t2, \$zero, 0xabcd

A resposta correcta é:

```
lui $t2, 0x1001  
ori $t2, $t2, 0x0018
```

Pregunta **7**

Completa

Puntuación: 1,00  
sobre 1,00

🚩 Marcar a  
pregunta

¿En qué dirección de memoria se almacena n4 ?

.data

n1: .word 8,5,9

n2: .ascii "Hi!"

n3: .space 4

n4: .word 2

teniendo en cuenta que la dirección inicial de almacenamiento en el segmento de datos es

0x10010000

Seleccione unha:

☐ a. 0x10010010

☐ b. 0x1001000f

☒ c. 0x10010014

☐ d. 0x1001000c

A resposta correcta é;  
0x10010014

Pregunta **8**  
 Non respondida  
 Puntúa como 1,00  
 Marcar a pregunta

El segmento de datos de un programa en ensamblador es como sigue:  
 .data  
 vector: .word 0x1001cdab

El formato de almacenamiento de datos de nuestro procesador que direcciona la memoria a nivel de byte es Little-Endian, ¿como se almacena en memoria esta palabra?  
 Nótese que la posición más alta de memoria en las posibles respuestas que se muestran a continuación es la situada más a la izquierda.

- Seleccione unha:
- ☐ a. 0xabcd0110
  - ☐ b. No se puede almacenar ya que no es un entero.
  - ☐ c. 0x1001cdab
  - ☐ d. 0xbadc1001

A resposta correcta é: 0x1001cdab

Pregunta **9**  
 Completa  
 Puntuación: 1,00 sobre 1,00  
 Marcar a pregunta

Data segment				
MEMORY				
[0x10010000]	0x00000001	0x00000002	0x00000003	0x706d6f43
[0x10010010]	0x6574736f	0x0000616c	0x00000006	0x000000c8
STACK				
[0x6ffffffa]			0x00000001	0x00000002
	0x00000003	0x706d6f43	0x00000006	0x000000c8

¿Qué dato hay almacenado en la posición 0x6ffffff4 de la pila?

- Seleccione unha:
- ☐ a. Esa posición no está siendo utilizada.
  - ☒ b. 0x706d6f43
  - ☐ c. 0x00000003
  - ☐ d. 0x00000006

A resposta correcta é:  
 0x706d6f43



A resposta correcta é:  
0x00400090