

Trong phần mềm Mars, cửa sổ Data Segment có thể hiển thị dữ liệu ở vùng nhớ nào?

Vùng nhớ chứa lệnh

Vùng nhớ chứa các thanh ghi

Vùng nhớ chứa các biến

Vùng nhớ ngăn xếp

Trong MIPS, khai báo nào sau đây là khai báo vùng lệnh

.data #khai báo biến

.stack

.code

.text #khai báo lệnh

Biết rằng vùng nhớ ngăn xếp (stack) có chiều từ đáy đến đỉnh ngược với chiều tăng của địa chỉ bộ nhớ chính. Để cất dữ liệu vào đỉnh stack cần sử dụng lệnh nào sau đây

A.

addi \$sp,\$sp,-4

sw \$s0,0(\$sp)

B.

addi \$sp,\$sp,0

sw \$s0,0(\$sp)

C.

addi \$sp,\$sp,4

sw \$s0,0(\$sp)

D.

addi \$sp,\$sp,-8

sw \$s0,4(\$sp)

Đoạn lệnh nào sau đây có xảy ra hiện tượng tràn số học (OVERFLOW)

A, li \$s0,0xFFFFFFFF

li \$s1,0x80000000

add \$s2,\$s0,\$s1

B, li \$s0,0xFFFFFFFF

li \$s1,0x80000000

sub \$s2,\$s0,\$s1

C, li \$s0,0xFFFFFFFF

li \$s1, 0xFFFFFFFF

add \$s2,\$s0,\$s1

D, li \$s0,0xFFFFFFFF

li \$s1, 0xFFFFFFFF

sub \$s2,\$s0,\$s1

0xFFFFFFFF = -1

0x80000000 = số âm bé nhất

Khai báo 1 biến như sau:

.data

X: .word 10

Hỏi biến X chiếm mấy byte bộ nhớ: **4 byte**

Lựa chọn nào là lệnh tương ứng với mã máy sau 0x21290001?

addi \$t0, \$t0, 1

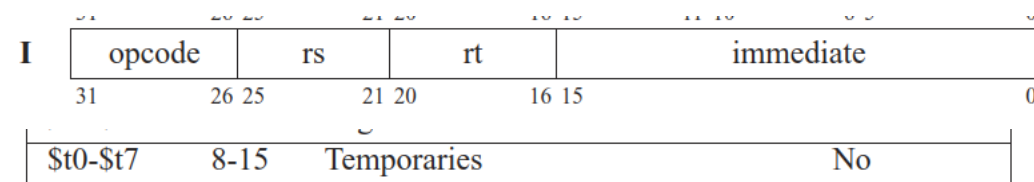
addi \$t1, \$t1, 1

addi \$t3, \$t3, 1

addi \$t2, \$t2, 1

0010 00|01 001|0 1001| 0000 0000 0000 0001

rs = rt - 9 => \$t1



Trong các khuôn dạng lệnh MIPS, trường opcode (mã thao tác) nằm ở vị trí nào?

Bit 0-5

Bit 26-31

Bit 21-25

Bit 16-20

Trong nhóm lệnh I-type của tập lệnh MIPS (không bao gồm giả lệnh), ví dụ lệnh addi \$t0, \$zero, imm thì giá trị lớn nhất của toán hạng imm là bao nhiêu (imm là giá trị có dấu)

$2^{15}-1$

$2^{32}-1$

2^8-1

$2^{16}-1$

Số bit dành cho imm = 16, imm có dấu

Câu lệnh sau khi biên dịch sẽ chiếm mấy byte bộ nhớ? **8**

la \$a0, 0x10203040

la là giả lệnh, được biên dịch thành 2 lệnh

Lệnh sau đây thuộc nhóm nào

addi \$s0, \$zero, 0x0000

Pseudo-instruction (Giả lệnh)

J-type

R-type

I-type

Add Immediate addi I $R[rt] = R[rs] + \text{SignExtImm}$ (1,2) 8_{hex}

Hãy dịch lệnh sau thành mã máy (mã máy viết dưới dạng hexa). Biết rằng lệnh add có các trường opcode = 0, funct = 32. Số hiệu thanh ghi t0 là 8.

add \$t0, \$t3, \$t4

rs = t3, rt = t4, rd = t0

Add add R $R[rd] = R[rs] + R[rt]$ (1) 0 / 20_{hex}

R	opcode	rs	rt	rd	shamt	funct
	31	26 25	21 20	16 15	11 10	6 5
						0

0000 00|01 011|0 1100| 010 0 0|000 00|01 0000|

0x016C4010

Chọn phát biểu đúng với chương trình sau đây thực hiện trên công cụ MARS

.eqv IN_ADRESS_HEXa_KEYBOARD 0xFFFF0012

.data

Message: .asciiz "Key is pressed\n"

.text

main:

li \$t1, IN_ADRESS_HEXa_KEYBOARD

li \$t3, 0x80

sb \$t3, 0(\$t1)

Loop: nop

nop

b Loop #polling

end_main:

.ktext 0x80000180

IntSR: addi \$v0, \$zero, 4

```

    la    $a0, Message
    syscall

next_pc:mfc0    $at, $14
    addi    $at, $at, 4
    mtc0    $at, $14

return: eret

```

Chương trình cài đặt cơ chế thăm dò (polling) khi có phím trên keyboard matrix được bấm sẽ hiện xâu thông báo ra màn hình

Chương trình cài đặt cơ chế ngắt khi có phím trên keyboard matrix được bấm thì sẽ thực hiện chương trình con xử lý ngắt tại địa chỉ 0x80000180

Chương trình cài đặt cơ chế ngắt trong đó chương trình chính sẽ gọi chương trình xử lý ngắt

Chương trình cài đặt cơ chế thăm dò (polling) trong đó chương trình chính sẽ gọi chương trình con tại địa chỉ 0x80000180

Chương trình sau thực hiện quét bàn phím ma trận (Keyboard matrix 4x4) trong công cụ MARS.

Chọn các phát biểu đúng với chương trình này

```

.equ IN_ADRESS_HEXА_KEYBOARD    0xFFFF0012
.equ OUT_ADRESS_HEXА_KEYBOARD    0xFFFF0014

.text

main:

    li    $t1, IN_ADRESS_HEXА_KEYBOARD
    li    $t2, OUT_ADRESS_HEXА_KEYBOARD

polling:

    li    $t3, 0x08    #chỉ số hàng dưới cùng
    sb    $t3, 0($t1)
    lb    $a0, 0($t2)

print:

    li    $v0, 34
    syscall

sleep:

    li    $a0, 100
    li    $v0, 32
    syscall

j    polling

```

Chương trình quét 1 hàng của bàn phím keyboard matrix bằng cách ghi chỉ số hàng tương ứng trong thanh ghi \$t3 vào địa chỉ 0xFFFF0014

Chương trình quét 1 hàng của bàn phím keyboard matrix bằng cách ghi chỉ số hàng tương ứng trong thanh ghi \$t3 vào địa chỉ 0xFFFF0012

Chương trình quét tất cả các hàng của bàn phím keyboard matrix bằng phương pháp polling

Chương trình chỉ quét 1 hàng cuối (row 4) của bàn phím keyboard matrix bằng phương pháp polling

#	col 0x1	col 0x2	col 0x4	col 0x8
#				
# row 0x1	0	1	2	3
#	0x11	0x21	0x41	0x81
#				
# row 0x2	4	5	6	7
#	0x12	0x22	0x42	0x82
#				
# row 0x4	8	9	a	b
#	0x14	0x24	0x44	0x84
#				
# row 0x8	c	d	e	f
#	0x18	0x28	0x48	0x88
#				

Giả sử có đoạn chương trình như sau:

.data

x: .word 2023 #0x000007E7

.text

la \$at, x

lb \$t0, 1(\$at)

Hỏi sau khi đoạn chương trình trên được thực hiện, thanh ghi \$t0 có giá trị là bao nhiêu? Biết rằng MARS cài đặt lưu trữ dữ liệu theo kiểu Little-Endian (byte thấp lưu ở ngăn nhớ có địa chỉ thấp, byte cao lưu ở ngăn nhớ có địa chỉ cao)

0x07

0x7e

0x00

0xe7

0xE7	0x00000000
0x07	0x00000001
0x00	0x00000002
0x00	0x00000003

Giá trị: đọc từ dưới lên

Lb: đọc từ trên xuống

Xác định giá trị thanh ghi \$a1 sau khi đoạn chương trình sau được thực hiện với công cụ MARS. Biết rằng công cụ MARS cài đặt thứ tự lưu trữ dữ liệu trong bộ nhớ theo kiểu little-endian (byte thấp LSB lưu ở địa chỉ thấp hơn)

.data

A0: .byte 0x01, 0x02, 0x03, 0x04 #lưu 4 dòng từ trên xuống

.text

la \$a0, A0

lw \$a1, (\$a0) #load 4 dòng từ dưới lên

0x01	0x00000000
0x02	0x00000001
0x03	0x00000002
0x04	0x00000003

0x01

0x01020304

0x04030201

0x04

Tại địa chỉ 0x10010000 khai báo biến .word có giá trị 0x01020304

Hỏi giá trị của byte nhớ tại địa chỉ 0x10010003 là bao nhiêu. Biết rằng phần mềm MARS cài đặt thứ tự lưu trữ trong bộ nhớ theo kiểu little-endian (byte thấp được lưu trữ ở ngăn nhớ có địa chỉ thấp, byte cao được lưu trữ ở ngăn nhớ có địa chỉ cao)

0x04	0x10010000
0x03	0x10010001
0x02	0x10010002
0x01	0x10010003

0x01

0x04

0x03

0x02

Biết rằng vùng DATA 0x10010000, trong đoạn mã sau thanh ghi \$t1 sẽ được load nội dung byte nhớ ở địa chỉ nào

.data

x: .word 0x01020304

.text

la \$t0,x

lb \$t1,2(\$t0)

0x04	0x10010000
0x03	0x10010001
0x02	<u>0x10010002</u>
0x01	0x10010003

0x10010002

0x10010003

0x10010001

0x10010000

Giả sử có đoạn chương trình như sau:

.data

x: .word 10

y: .word 11

z: .word 12

.text

la \$t0, x

la \$t1, y

la \$t2, z

Sau khi đoạn chương trình trên được thực hiện, thanh ghi \$t1 có giá trị là 0x10010004. Hỏi thanh ghi \$t2 có giá trị là bao nhiêu (điền số dưới dạng hexa)?

0x10010008

X	10	0x10010004
Y	11	0x10010008
z	12	0x1001000C

Cho biết giá trị của thanh ghi \$s1 trong hệ 10 sau khi đoạn chương trình sau được thực hiện

.text

addi \$s0, \$zero, 0xA #s0 = 0x0000000A 4 bit cuối = 1010

and \$s1, \$s0, 0x7 #0x07 4 bit cuối = 0111

\$s1 = 1010 and 0111 = 0010 = 2

Để thực hiện đảo các bit của byte thấp nhất (LSB) của thanh ghi \$s0 ta dùng lệnh nào sau đây, biết giá trị \$t0=0xFF,\$t1=0x00

AND \$s0,\$s0,\$t0

XOR \$s0,\$s0,\$t0

OR \$s0,\$s0,\$t0

AND \$s0,\$s0,\$t1

Đảo bit là phép so với 1

Để thực hiện xóa các bit của byte thấp nhất (LSB) của thanh ghi \$s0 ta dùng lệnh nào sau đây, biết giá trị \$t0=0xFF,\$t1=0xFFFFF00

OR \$s0,\$s0,\$t0

AND \$s0,\$s0,\$t1

XOR \$s0,\$s0,\$t0

AND \$s0,\$s0,\$t0

Xoá là AND với 0

Những lệnh nào sau đây có thể được sử dụng để xóa nội dung thanh ghi \$s0

AND \$s0, \$s0, \$zero

XOR \$s0, \$s0, \$s0

XOR \$s0, \$s0, \$zero

AND \$s0, \$s0, \$s0

Xoá là AND với 0

Cho biết giá trị của thanh ghi s1 sau khi thực hiện đoạn lệnh sau

li \$s0,5 #s0 = 5 = 0x00000005, 4 bit cuối = 0101

srl \$s1,\$s0,1 #dịch phải 1 bit, 4 bit cuối = 0010

sll \$s1,\$s1,1 #dịch trái 1 bit, 4 bit cuối = 0100 = 4

Đoạn chương trình sau hiển thị gì ra màn hình Run I/O của công cụ MARS:

.data

str1: .ascii "Hello" #hàm ascii ko có "z" => ko có \0 => đọc tiếp

str2: .asciiz "World"

```
.text
    li $v0, 4
    la $a0, str1
    syscall
World
Hello
HelloWorld
Hello World
```

Đoạn chương trình sau hiển thị chuỗi ký tự gì ra màn hình Run I/O:

```
.data
    str: .asciiz "hello"
.text
    la $a0, str
    li $a1, 0
label1:
    add $a2, $a1, $a0
    lb $a3, ($a2)
    beq $a3, $zero, label2
    add $a3, $a3, -32 #UPPERCASE
    sb $a3, ($a2)
    addi $a1, $a1, 1
    j label1
label2:
    li $v0, 4
    syscall
```

HELLO

Chọn phát biểu đúng khi thực hiện chương trình sau:

```
.text
    addi $t0, $zero, 1 #t0 = 1
    jal f1
    addi $t0, $zero, 2
f1:
    addi $t0, $zero, 3 #t0 = 4
    jal f2
    addi $t0, $zero, 4 #t0 = 13, $ra tại vị trí này vẫn đang là t0 + 4
    jr $ra
```


f2:

```
addi $t0, $zero, 5 #t0 = 5 $ra tại đây là sau jal f2 là t0 + 4
```

```
jr $ra
```

Sau khi chương trình kết thúc thanh ghi \$t0 có giá trị bằng 2

Sau khi chương trình kết thúc thanh ghi \$t0 có giá trị bằng 5

Sau khi chương trình kết thúc thanh ghi \$t0 có giá trị bằng 1

Chương trình bị lặp vô hạn, không dừng.

Sau khi chương trình kết thúc thanh ghi \$t0 có giá trị bằng 4

Sau khi chương trình kết thúc thanh ghi \$t0 có giá trị bằng 3

Cho đoạn lệnh sau:

main:

```
li $a0,-45
```

```
jal func #nhảy đến func, $ra = địa chỉ lệnh nop
```

```
nop #thực hiện lệnh tại địa chỉ $ra => $pc = $ra
```

endmain:

func:

```
...
```

```
jr $ra
```

Hỏi sau thực hiện sau câu lệnh jr \$ra, thanh ghi PC (Program Counter) sẽ chứa địa chỉ của lệnh nào tiếp theo

li \$a0,-45

nop

jr \$ra

jal func

Giá trị của thanh ghi \$s0 sau khi đoạn chương trình sau được thực hiện (Nhập giá trị thập phân):

.text

```
addi $s0, $zero, 0 #s0 = 0
```

```
addi $s1, $zero, 1 #s1 = 1
```

```
addi $s2, $zero, 10 #s2 = 10
```

loop:

```
slt $t0,$s2,$s1
```

```
bne $t0,$zero,end #chương trình kết thúc khi $s2 < $s1
```

```
addi $s0,$s0,1 #s0 = s0 + 1
```

```
addi $s1, $s1, 1 #s1 = s1 + 1
```

```
addi $s2, $s2, -1 #s2 = s2 - 1
```

```
j loop
```

end:

Loop	S1	S2	S0
0	1	10	0
1	2	9	1
2	3	8	2
3	4	7	3
4	5	6	4
5	6	5	5
6			

\$s0	16	0x00000005
\$s1	17	0x00000006
\$s2	18	0x00000005

\$s0 = 5

Giá trị thanh ghi hi sau khi thực hiện đoạn lệnh sau là bao nhiêu (**\$hi lưu phần dư**)

addi \$t1, \$zero, -7 #t1 = -7

addi \$t2, \$zero, 2 #t2 = 3

div \$s0, \$t1, \$t2 #s0 = t1/t2

0x00000001

0xFFFFFFFF

Báo lỗi

0xFFFFFFF0

-7/2 = -3

-3 * 2 = -6, cộng thêm **-1** thành -7

Giá trị thanh ghi lo sau khi thực hiện đoạn lệnh sau là bao nhiêu (**\$lo lưu phần thương**)

addi \$t1, \$zero, -7

addi \$t2, \$zero, 2

div \$s0, \$t1, \$t2

0x00000001

0xFFFFFFF0

Báo lỗi

0xFFFFFFFF