**Computer Science**

**C.Sc. 342**

**Quiz No.2 To be performed**

**5:00-6:15 PM on**  March 23, 2022

Submit by 6:15 PM 03/23/2022on Slack to Instructor **Please write your Last Name on every page:**

**NO CORRECTIONS ARE ALLOWED IN ANSWER CELLS!!!!!**

You may use the back page for computations.

Please answer all questions. **Not all questions are of equal difficulty.**

**Please review the entire quiz first and then budget your time carefully.**

Please hand write and sign statements affirming that you will not cheat:

*“I will neither give nor receive unauthorized assistance on this exam. I will use only one computing device to perform this test”*

Please hand write and sign here:

This quiz has 6 pages.

|  |  |  |
| --- | --- | --- |
| Question | Your Grade | Max Grade |
| 1.1 |  | 5 |
| 1.2 |  | 10 |
| 1.3 |  | 10 |
| 1.4 |  | 10 |
| 2.1.1 |  | 15 |
| 2.1.2 |  | 15 |
| 2.1.3 |  | 15 |
| 2.2.1 |  | 5 |
| 2.2.2 |  | 5 |
| 2.2.3 |  | 5 |
| 2.3 |  | 5 |
|  |  |  |
|  |  |  |

Total: 100

**Question 1.**

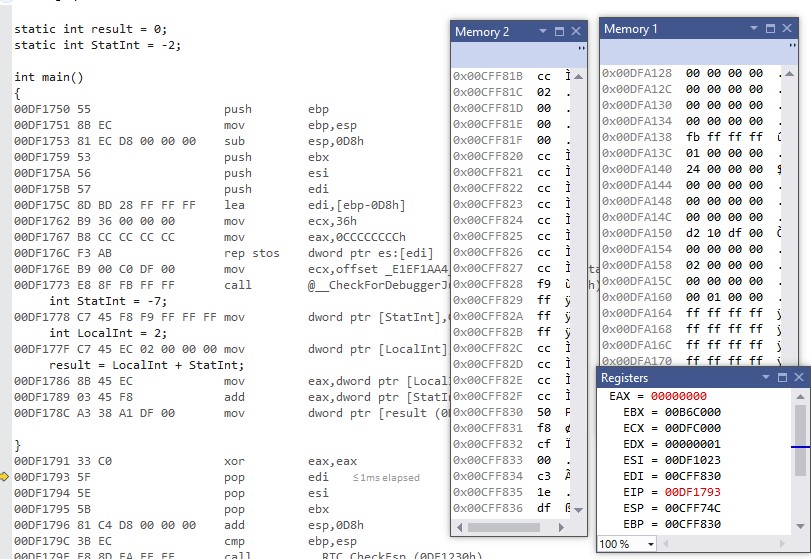
A student, while debugging his program, unintentionally displayed partially corrupted DISSASSEMBLY windows in MS Visual Studio Debug environment.

He was able to display correctly Register window, and two Memory windows.

His task was to determine addresses of variables in the expression **result = LocalInt + StatInt** in Memory at the instance of the snapshot. He is not allowed to restart the debug session.

Can you help him to answer the following

questions:



**1.1** [5 points] What is the address of the instruction that will be executed next instance?

**ANSWER:** This is shown on the EIP register. The EIP register contains the value. If we look closely at EIP register, it shown that the address is 0x00DF1793.

**1.2** [10 points] Can you determine the address of variable **StatInt** in the expression? **YES or NO.**

*Please circle around your answer***. IF** *No is your answer, then go to the next question* **ELSE** *Please compute the address of variable* ***StatInt*** *in memory , and determine the value of variable StatInt you can read from memory:*

*Address of* ***StatInt*** *is ……*0x00CFF828*.*

*Value of* ***StatInt*** *in memory is* 0x FF FF FF F9 = -7

*Please justify your answers.*

**ANSWER:**

The address of the variable StatInt is 0x00CFF830 + F8(offset) = 0x00CFF828.

**Reason:** The value of -7 in hexadecimal is FFF9, therefore, we can see in the memory windows 2 that the value of F9 FF FF is at address 0x00CFF828.

**1.3** [10points] Can you determine the address of variable **LocalInt** in the expression? **YES or NO.**

*Please circle around your answer***. IF** *No is your answer, then go to the next question* **ELSE** *Please compute the address of variable* ***LocalInt*** *in memory , and determine the value of variable* ***LocalInt*** *you can read from memory:*

*Address of* ***LocalInt*** *is …….* 0x00CFF81C

*Value of* ***LocalInt*** *in memory is….* 0x 00 00 00 02 = 2

*Please justify your answers.*

**ANSWER:**

The memory address is 0x00CFF830 + EC(offset) = 0x00CFF81C.

**Reason:** The value of 2 in hexadecimal is 0002 and if we look at the memory window 2, it holds the value of 2 in hexadecimal.

**1.4** [10 points] Can you determine the address of variable **result** in the expression? **YES or NO.**

*Please circle around your answer***. IF** *No is your answer, then go to the next question* **ELSE** *Please compute the address of variable* **result** *in memory , and determine the value of variable* **result** *you can read from memory:*

*Address of* **result** *is …….*0x00DFA138

*Value of* **result** *in memory is* FFFFFFFB *Please justify your answers.*

**ANSWER:**

The memory address is 0x00DFA138.

**Reason:** The value of -5 in hexadecimal is FFFFFFFB and if we look at the memory window 1, it holds value of -5 in hexadecimal.

**Question 2.**

*A student wrote MIPS assembly program and executed it in MARS simulator.*

.data

array1: .word -1,0x7fffffff,0x10000080,0x80000010

.text

main:

la $t1,array1

# create Frame pointer

add $fp,$zero,$sp

#Store the address of the first element on stack using frame pointer

sw $t1,0($fp) #allocate memory on Stack for 6 integers

addi $sp,$sp,-24

#load **FIRST** element from array1[0] to register $s0

lw $s0,0($t1)

#**push $s0 (NO PUSH!)**i.e. store register $s0 on #top of the stack  sw $s0,0($sp)

#load **SECOND** element from array1[1] to register $s0

lw $s0,4($t1) #create new top of the stack  addi $sp,$sp,-4  sw $s0,0($sp)

#

#load third element from array1[2] to register $s0  lw $s0,8($t1) #create new top of the stack

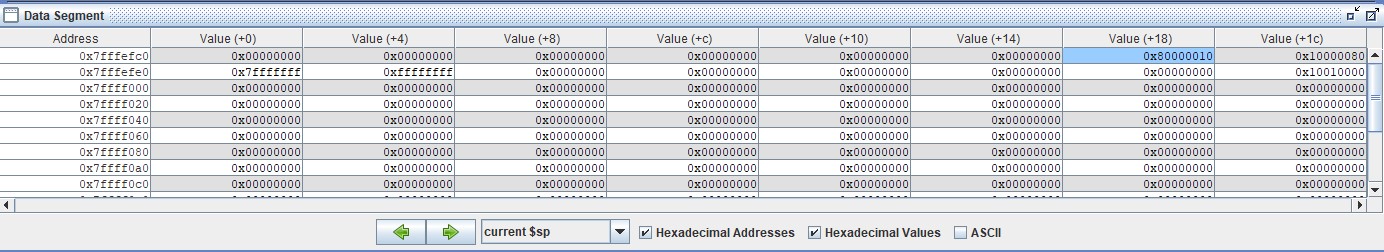
# addi $sp,$sp,-4 sw $s0,0(sp)

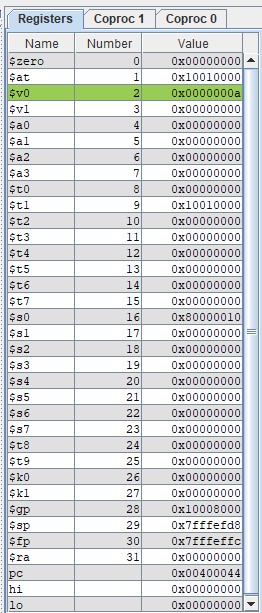
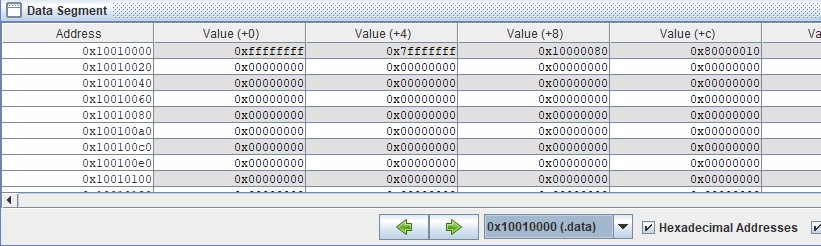
#load forth element from array1[3] to register $s0

lw $s0,12($t1)

# #create new top of the stack addi $sp,$sp,-4 sw $s0,0($sp)

After execution of the program in MARS simulator, he displayed the following memory windows and register file:





**Figure 2. Register file and memory windows in MARS simulator.**

Based on the information displayed in **Figure 2.** memory windows and register file above, please answer the following questions

2.1.1 [15 points] What is the address of an integer that was **first** pushed on to stack?

**ANSWER:**

The address of an integer that was first pushed on to stack is 0x7fffefe4 + 0x4(offset) = 0x7fffefe4 because the value in $fp in the $sp value before inserting and the value of $sp decrease by -24.

2.1.2 [15 points] What is the value in Hex and signed decimal of an integer that was **first** pushed on to stack?

**ANSWER:**

The value in hex of the integer is 0xffffffff. This is in signed decimal the equivalent of -1. The signed decimal can be calculated by using 2’s complement.

2.1.3 [15 points] What is the offset from FRAME POINTER to an integer that was **first** pushed on to stack?

**ANSWER:**

The offset from FRAME POINTER to an integer that was **first** pushed on to stack is the same as the frame pointer.

This means, 0x7FFFEFFC – 0x7FFFEFE4 = 0x18. The answer can be 0xE8 which means -24.

2.2.1 [5 points] What is the address of an integer that was **Last** pushed on to stack?

**ANSWER:**

The address of the last value pushed on the stack is 0x7fffefc0 + 0x18(-24 offset in hex). This equals to 0x7fffeffd8.

Therefore, the address of the last value on pushed on stack is 0x7fffeffd8.

2.2.2 [5 points] What is the value in Hex and signed decimal of an integer that was **Last** pushed on to stack?

**ANSWER:**

The hex value of the last value pushed on the stack is 0x80000010 and the decimal signed value is -8\*16^7 + 1\*16^1 = -2147483632.

2.2.3 [5 points] What is the offset from FRAME POINTER to an integer that was **Last** pushed on to stack?

**ANSWER:**

The offset from the frame pointer of the integer that was last pushed on the stack is 0x7FFFEFFC– 0x7fffefd8= -36 in decimal.

2.3 [5 points] Based on the data shown Figure 2.,Can you determine if Frame pointer points to an **address** *or a* **value?** Please circle around your answer. Please explain.

**ANSWER:**

The frame pointer points to the starting address of the stack. It defers from the stack which has the same value as frame pointer. However, it changes as things are pushed onto the stack. In figure 2, we can see that the address of the frame pointer is 0x7FFFEFFC.