

Recitation 11

In this recitation you will explore an example MIPS program to learn more about the stack, activation records, and the calling convention that we are using in this class. To complete the lab answer the questions on this page. If you have any questions feel free to ask the TA.

1) Look at figure 1 in the attached handout, this shows the actual program stack immediately before main calls fact(3). What is the 3 on this stack? Is this correct so far?

**3 is the first argument to the fact function that main is about to call.
This is correct so far.**

2) Figure 2 in the attached handout shows the program stack immediately after the prologue of the fact(3) call. Recall that MIPS automatically sets up the frame pointer and stack pointer at the beginning of the program. Knowing this, and looking out the output in figure 2, does main have a frame pointer?

Looking at Fig 2 we can see that the old_fp (which contains the fp for main) is set to 0x00000000. The frame pointer is supposed to be a pointer into the stack. There is no way that 0x00000000 can be a pointer to the stack so it must be the case that MIPS does NOT set up a frame pointer for main.

3) In Figure 3 of the handout, draw an arrow pointing to where the stack pointer points and another arrow pointing to where the frame pointer points. Make sure to label the arrows clearly.

**The stack pointer is pointing to <1,arg1> (top of stack)
The frame pointer is pointing to <2,local1> (first local variable)**

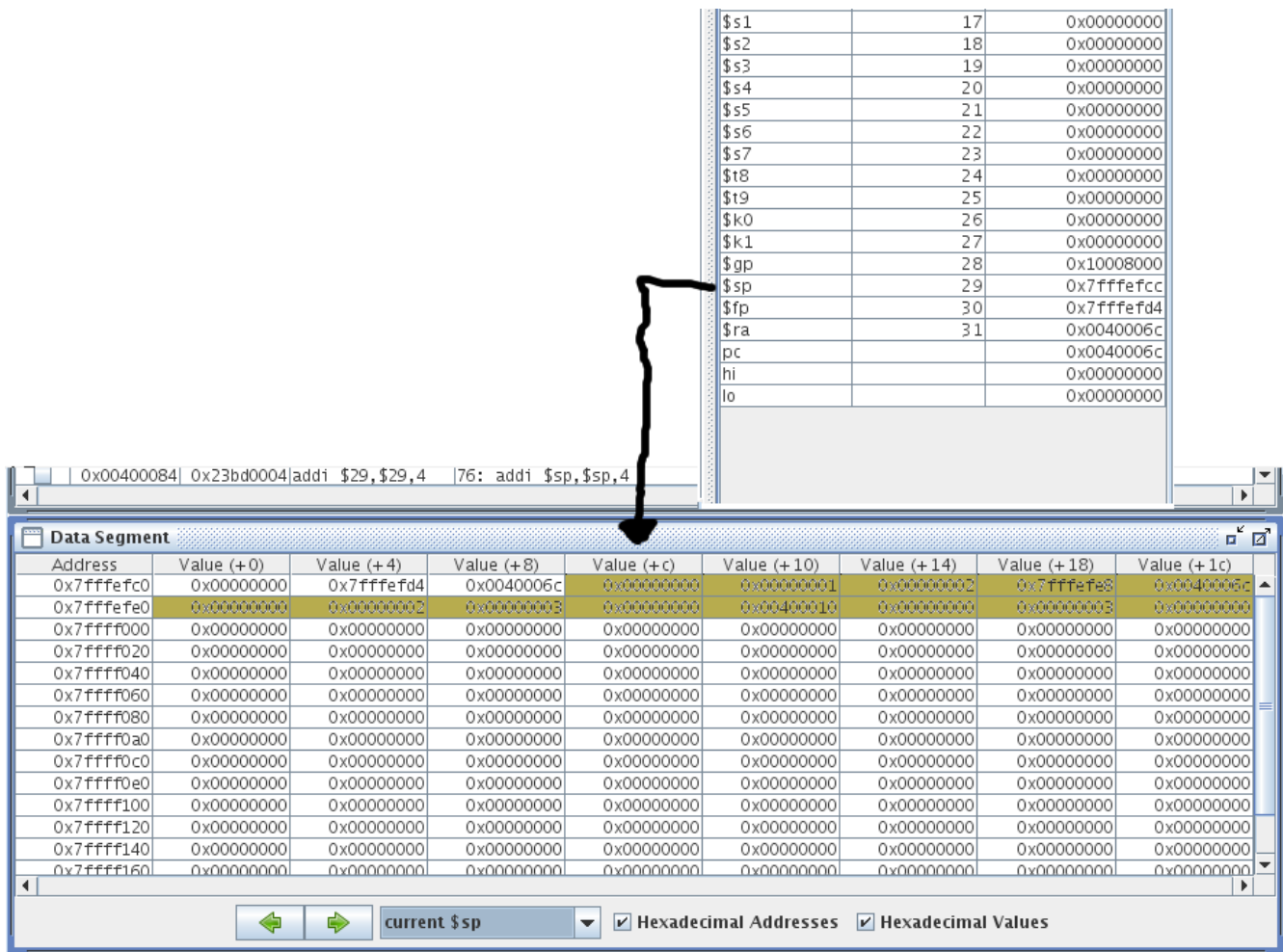
4) In Figure 3 circle the stack position corresponding to memory location x7FFFEFE8

To figure this out we first notice that x7FFFEFE8 is in our stack in the old_fp position for the activation record for fact(2). That means that x7FFFEFE8 is the frame pointer for fact(2). The frame pointer for stack 2 should be pointing to <3,local1>. This means that x7FFFEFE8 must be the address of the location <3,local1>.

The position <3,local1> should be circled.

Answers to 5 & 6

Theoretical stack before fact(1) calls epilogue (common incorrect answer for question 5)		Theoretical stack immediately after fact(1) returns (after epilogue) ANSWER TO QUESTION 5	Actual stack immediately after fact(1) returns (after epilogue) ANSWER TO QUESTION 6
1	local1	1	retval
x7FF...	old_fp	1	arg1
x004...	retaddr	2	local1
1	retval	x7FF...	old_fp
1	arg1	x004...	retaddr
2	local1		retval
x7FF...	old_fp	2	arg1
x004...	retaddr	3	local1
	retval	x000...	old_fp
2	arg1	x004...	retaddr
3	local1		retval
x000...	old_fp	2	arg1
x004...	retaddr	3	local1
	retval	x000...	old_fp
3	arg1	x004...	retaddr
			retval
		3	arg1



Doctored screenshot showing the actual stack (from MARS)

7) What is the bug & how would you fix it?

Line 73 in fact.mips (the first line of the epilogue) puts the return value into the appropriate position on the stack. This line assumes that the return value is in \$v0

In the recursive case this assumption is fulfilled by line 68. (which puts $n * \text{fact}(n-1)$ in \$v0)

In the base case we should put 1 into \$v0 but we never do. Thus fact(1) is returning 0.