EMBSYS100 - AU19
ASSIGNMENT 05

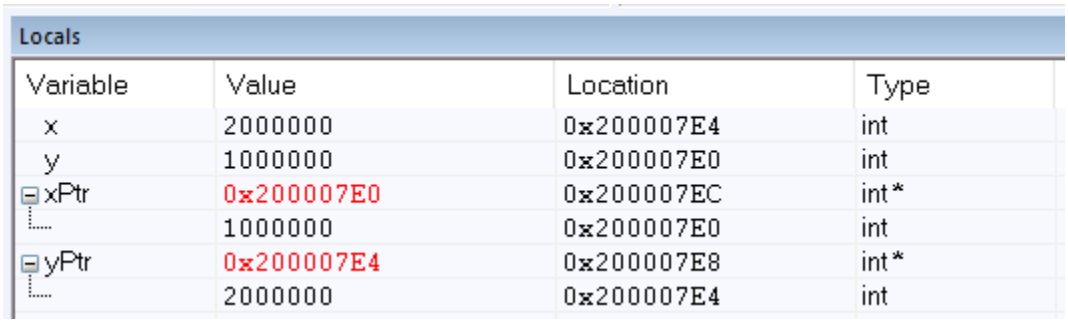
Goal

The goals for the assignment this week:

1. To gain in depth understanding of the pass-by-value nature of C.
2. To gain hands on experience with the ARM Cortex-M4 assembly language.

Problems:

1. Create a function that allows swapping of two pointers.
    a. Explain what the “main” function does to the CSTACK in order setup the input arguments prior to calling the swap_pointer() function?
    b. And what are the values in R0 & R1 when swap_pointer() is called?
    c. Share a screen shot of the local variables inside of “main” after the function swap_pointer() returns, showing the values of the pointers and what they are pointing to, similar to the picture below:



Locals

Variable	Value	Location	Type
x	2000000	0x200007E4	int
y	1000000	0x200007E0	int
⊟ xPtr	0x200007E0	0x200007EC	int *
└	1000000	0x200007E0	int
⊟ yPtr	0x200007E4	0x200007E8	int *
└	2000000	0x200007E4	int

2. Run **sqrAsm** code from the class demo:
    a. Copy HelloWorld from Assignment 1.
    b. Download the file sqrAsm.s and copy it to the project directory.
    c. In IAR, add sqrAsm.s file to your project. Now you should be able to see the file from the Workspace Window.
    d. Modify main.c to call sqrAsm function.
    e. Run your program on the board, and capture a snapshot image of the output from TeraTerm.
3. Create a new file **div2Asm.s** and add the file to the same HelloWorld project above.
    a. Write the assembly code to take an input argument, divide it by 2, and return the result.
    b. Invoke the function “PrintString” from within divAsm before doing the division computation.
    c. Add a comment for every statement in your assembly function code.
    d. In the main.c, invoke dev2Asm(foo) and capture the screen output from TeraTerm.

4. Implement a swap function in assembly and call it "**swapCharsAsm**":
    a. It takes as input two variables of char data type and swaps the two chars
    b. Add a comment for every statement in your assembly function code.
    c. **Bonus:** Returns 0 if the two chars are identical; otherwise, return 1.

5. **Bonus:** Implement the swap_pointer() function from #1 above in assembly and call it **swapPointersAsm**().
    a. Add a comment for every statement in your assembly function code.
    b. Take a snap shot of the output after invoking the swapPointerAsm() subroutine.

What to turn in and how:

- Check in all your homework in your repo under the folder "**assignment05**".
- Your folder should contain the following:
    - Turn in your source code files only (for example: main.c, …etc.) and any other files that you have authored.
    - Turn in answers to questions in markdown file format.
- Submit a link to your GitHub repo assignment:
    - Ex: "https://github.com/<account_id>/embsys100/assignment05"