Steven Reeves

CST 250

5/5/18

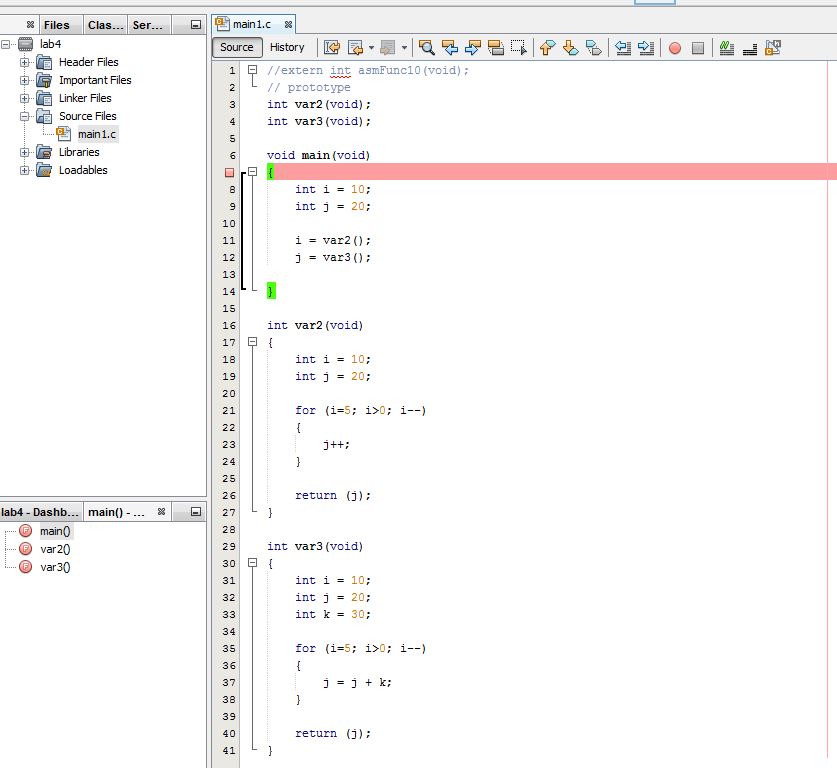
Lab 4 Report

**Introduction:**

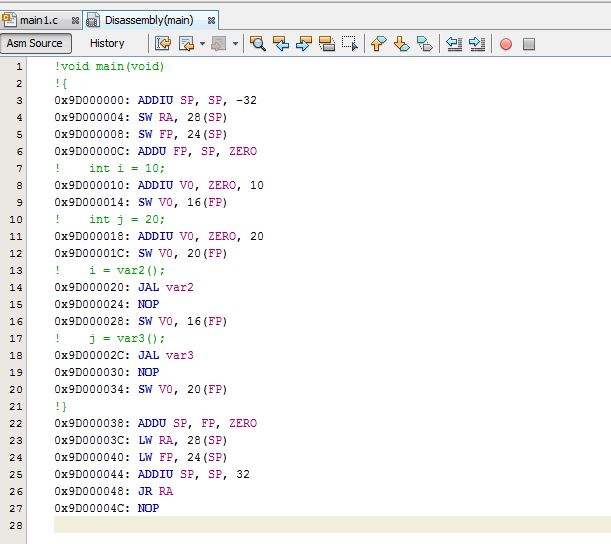
The objective of this lab is to use the disassembly view in the MPLAB X IDE to inspect stack frames as they’re getting used and created.

**Part 1:**

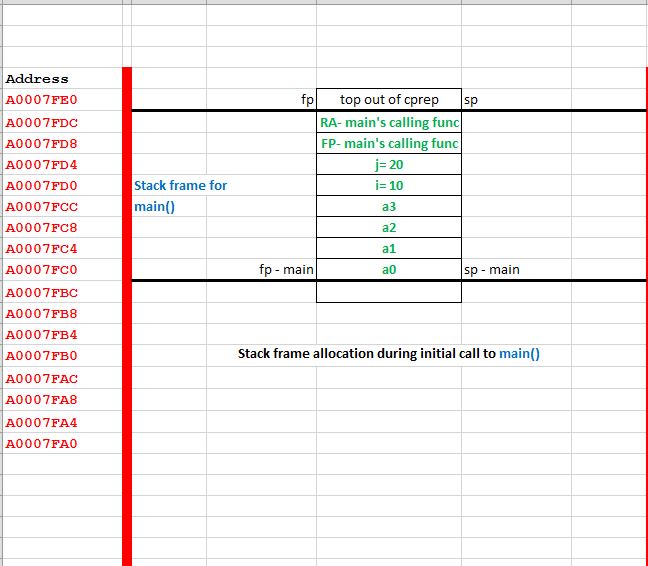
The first part of this lab had me create a new .c project using the same setup we have for previous labs. After dropping a breakpoint at the beginning of the main block, I saw the following:



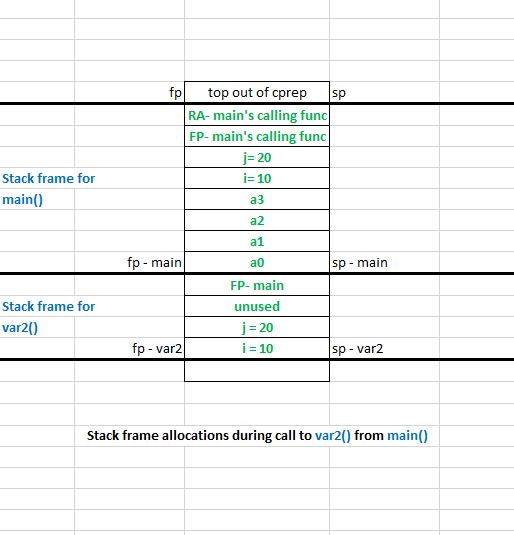
After starting a debugging session, Window->Debugging->Disassembly was used to access the following view:



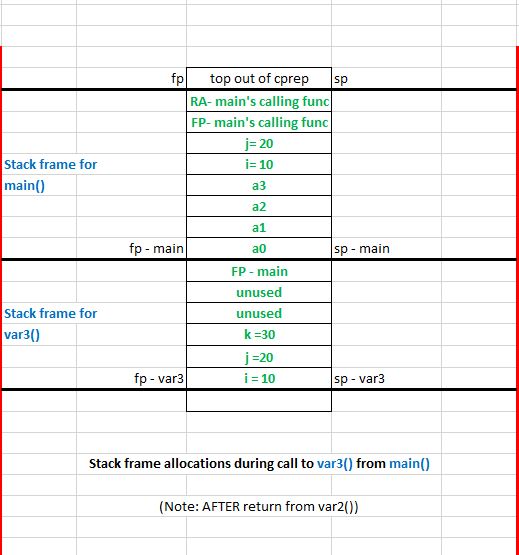
At this point, I was able to tell how the stack frame was constructed, and what was in it. Entering that information into the “Lab 4 Stack Frame Template” is shown below:



I then deleted my breakpoint from main and added one at the beginning of var2(). The same process of observing the disassembly view was used to obtain the following information:



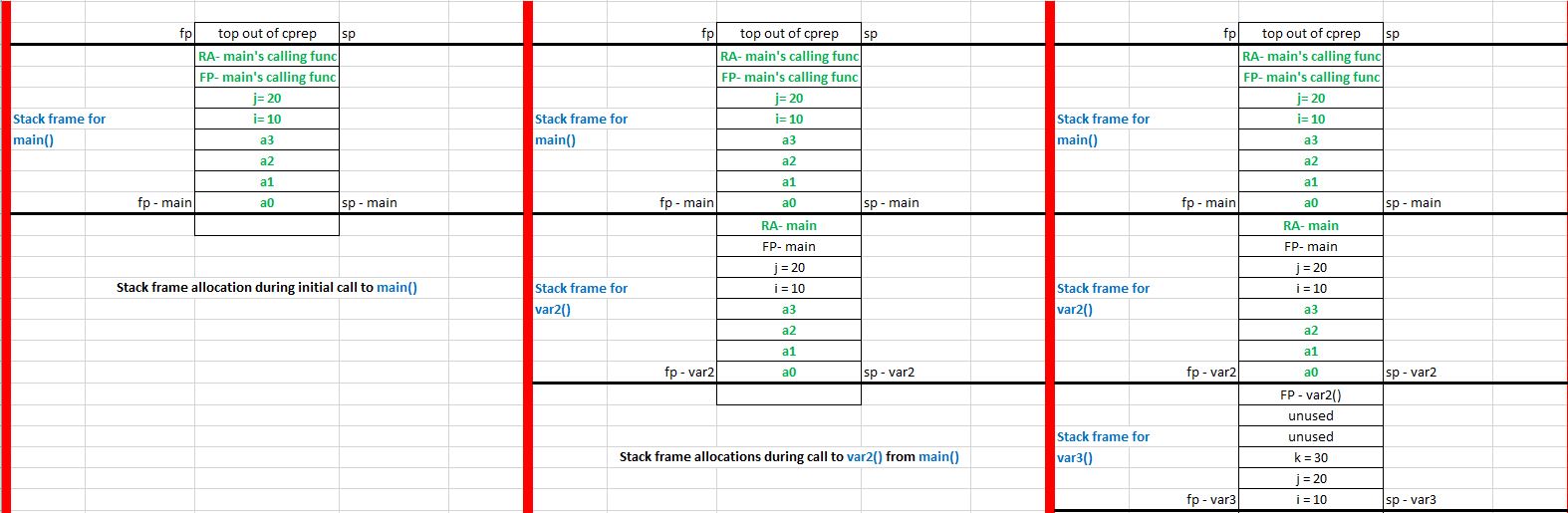
I then deleted my breakpoint from main and added one at the beginning of var3(). The same process of observing the disassembly view was used to obtain the following information:

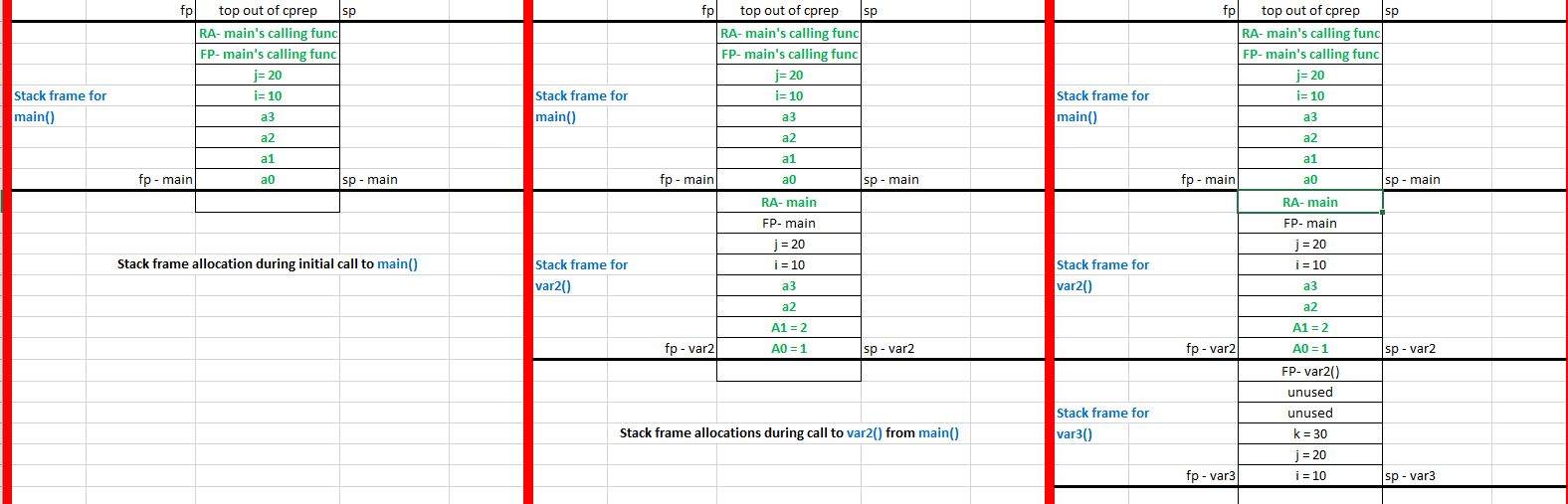


Note: These screenshots were created before asking Questions 1 and 2.

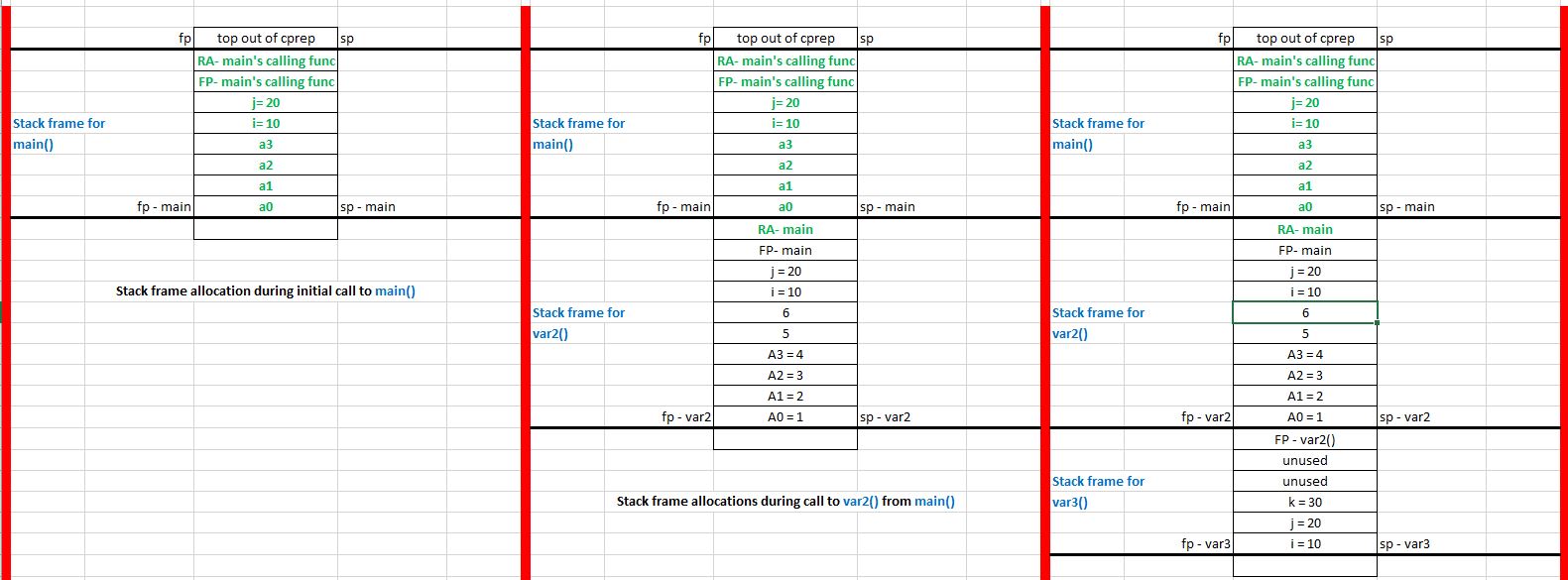
**Part 2:**

This part had me do the same process as above for the remaining main.c projects. Screenshots of the filled out Stack Frame Template are below:

main2.c:

main3.c:

main4.c:



**Conclusion:**

This lab was very difficult for me, as I missed the lecture before it. Here are some specific issues I had and how I figured them out.

1) Typo in “Lab 04.docx”: “Fill out the rows for what the memory locations in the stack from is being built to contain…”

- Assumed that “from” should be “frame”

2) Example “Lab 4 Stack Frame Template” posted from missed lecture was blank.

- This just took some time to figure out without an example

3) Having to restart the IDE every time I add the new main.c file

- Didn’t realize I had to do this until a couple of failures to run. Also, removing the breakpoints from previous main.c files helped.

4) Wasn’t sure if I was supposed to step through the code and evaluate variables as they changed

- Seeing that wasn’t mentioned in the lab documentation helped.

5) Not sure why RA isn’t used for the last function.

- <ANSWER>

6) I’m still a bit unclear on how a value is returned. It seems as if all the return values are just in V0. Interested to see how the stack frame comes into play with that.

7) Not having a working example was the hardest part for me in this.

- Powerpoint presentations helped a lot.

Overall, quite a difficult lab resulted in me just walking through the disassembly code very carefully and realizing how stack frames are used with different functions.

3) Question: Are we supposed to step through the function to see what is at the end?

4) Question : Why aren’t we storing a Return Address at “top” of stack frame? (because no return params?)