CSE301 Computer Architecture
Fall 2020
Programming Assignment #1

1. **Overview**: In this programming assignment, you will implement the quick sort algorithm using the MIPS assembly language. Please follow the steps below.

   - Refer to the quick sort code written in the C programming language, which we have provided.
   - Translate the quick_sort and partition functions written in C to the MIPS assembly language by exactly following the MIPS calling conventions.
   - Test (and debug) your assembly code using the SPIM simulator.

2. **Deliverables**

   - The assembly file (quick_sort.s) that implements the quick sort algorithm.
   - A screenshot that clearly shows the final output (e.g., the values stored in the memory) after successfully executing your assembly program in the SPIM simulator.

3. **How to submit**

   - Submit your deliverables through BB.
   - **Due: 5:00pm, November 6, 2020**

4. **Grading (tentative)**

   - Correctness: 60%
     - We may test your code with input data sets different from the ones provided in the skeleton code.
   - Code quality: 40%
     - For example, we will check if your code exactly follows the MIPS calling conventions.

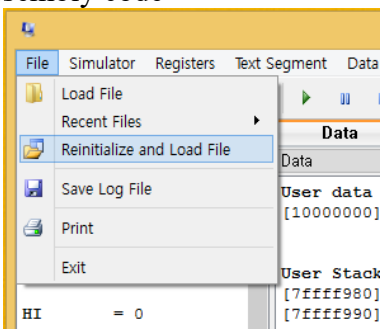
5. **How to test your assembly code using the SPIM simulator**

   - How to download and install the SPIM simulator
     - http://sourceforge.net/projects/spimsimulator/files/
       go to the above link and download file according to your OS.
       - Be sure to download the version 9.1.12 on Windows. Otherwise, you will get errors.
       - For MAC users, please try the version 9.1.13 and let us know if you encounter any issue.
     - Install normally

   - How to load and edit your assembly file

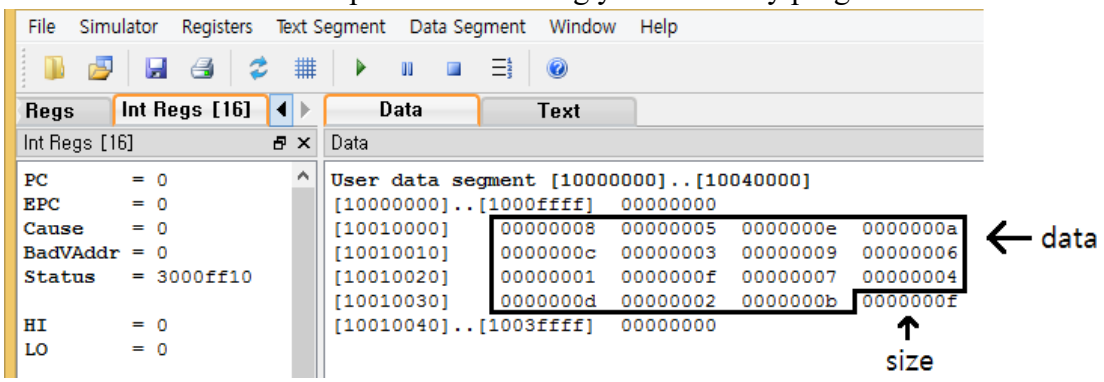
- Use notepad to write and edit assembly file

- How to execute your assembly code



  - Open up QtSpim and go to File -> Reinitialize and Load File
  - Select file you want to run and press Open (in this example, we will use the quick_sort.s file we have provided).
  - Press Run/Continue button (play button) to execute assembly code until end
  - Repeat the above steps to run assembly code again

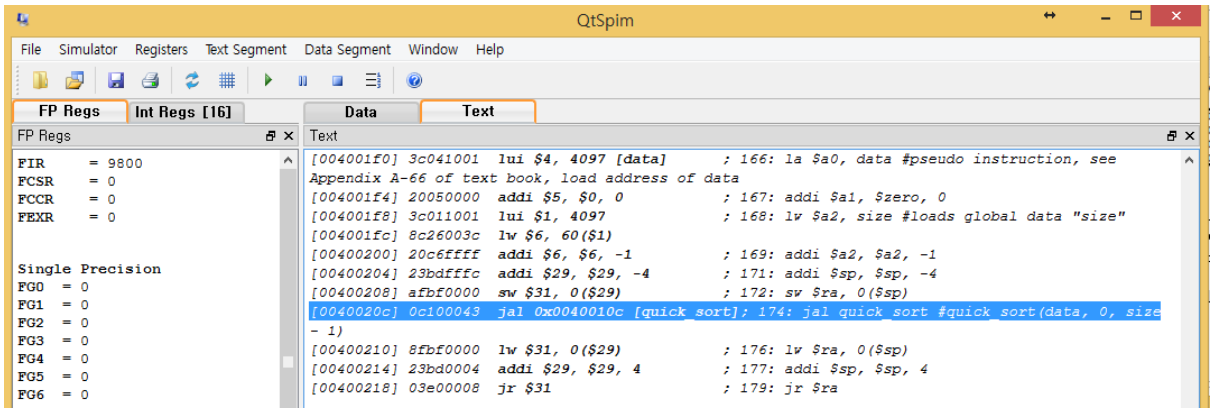
- How to check the output after executing your assembly program



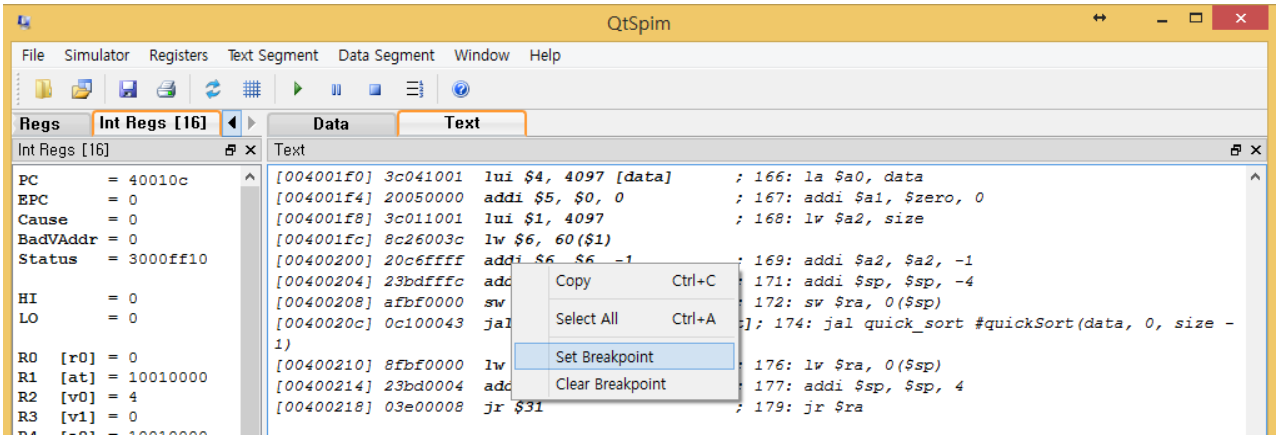
  - Left column will be about registers. Make sure it is set to “Int Regs [16]” tab
  - On the right column, switch to “Data” tab to see Data segment.
  - If quick_sort.s file is loaded successfully, the first n variables will indicate “data” array and last variable will indicate “size” variable in quick_sort.s.
  - If the sorting ran successfully, the first n variable will be in ascending order

- How to debug your assembly code

  - Press button in a rectangle to run instructions in a single step



  - On the right column, “Text” tab will highlight the next instruction to be executed.



  - To set a breakpoint, go to the “Text” tab. Right click on instruction you want to set breakpoint and click “Set Breakpoint”
  - If Run button is pressed, the program will stop when it reaches a breakpoint
  - Right click instruction and click “Clear Breakpoint” to remove the breakpoint