CS/EE 320 Computer Organization and Assembly Language (Spring 2025)

### Electrical Engineering Department

### SBASSE LUMS

### Instructor: Dr. Shahid Masud

### TA: Ibraheem

Note: QtSpim and VS Code are already installed in the lab systems. But if you want to download and install these software tools on your systems or laptops you can follow the following downloading and installation procedures.

# Introduction:

QtSpim is MIPS32 Simulator that reads and executes MIPS32 Assembly Language programs and displays the processor’s registers and memory. It cannot run the binary (compiled) code. QtSpim provides a simple debugger and small set of operating system services.

It can run on multiple platforms like Windows, Mac OS X, and Linux.

QtSpim does not have built-in text editor. Therefore, it is needed to download and install any suitable text editor. There are well known text editors available such as Notepad++, Atom, VS code, etc. We are using VS Code in this course. If you want to use your favorite text editor, then feel free to use it.

# Downloading:

QtSpim can be easily downloaded from the following link.

<https://sourceforge.net/projects/spimsimulator/>

VS code can be downloaded from the following link.

<https://code.visualstudio.com/download>

# Installation:

* Double click the downloaded file.
* Click “NEXT” when the installation window pops up.
* Agree the terms & conditions and select the location where you want to install QtSpim.
* After Installation is completed, click “FINISHED”.

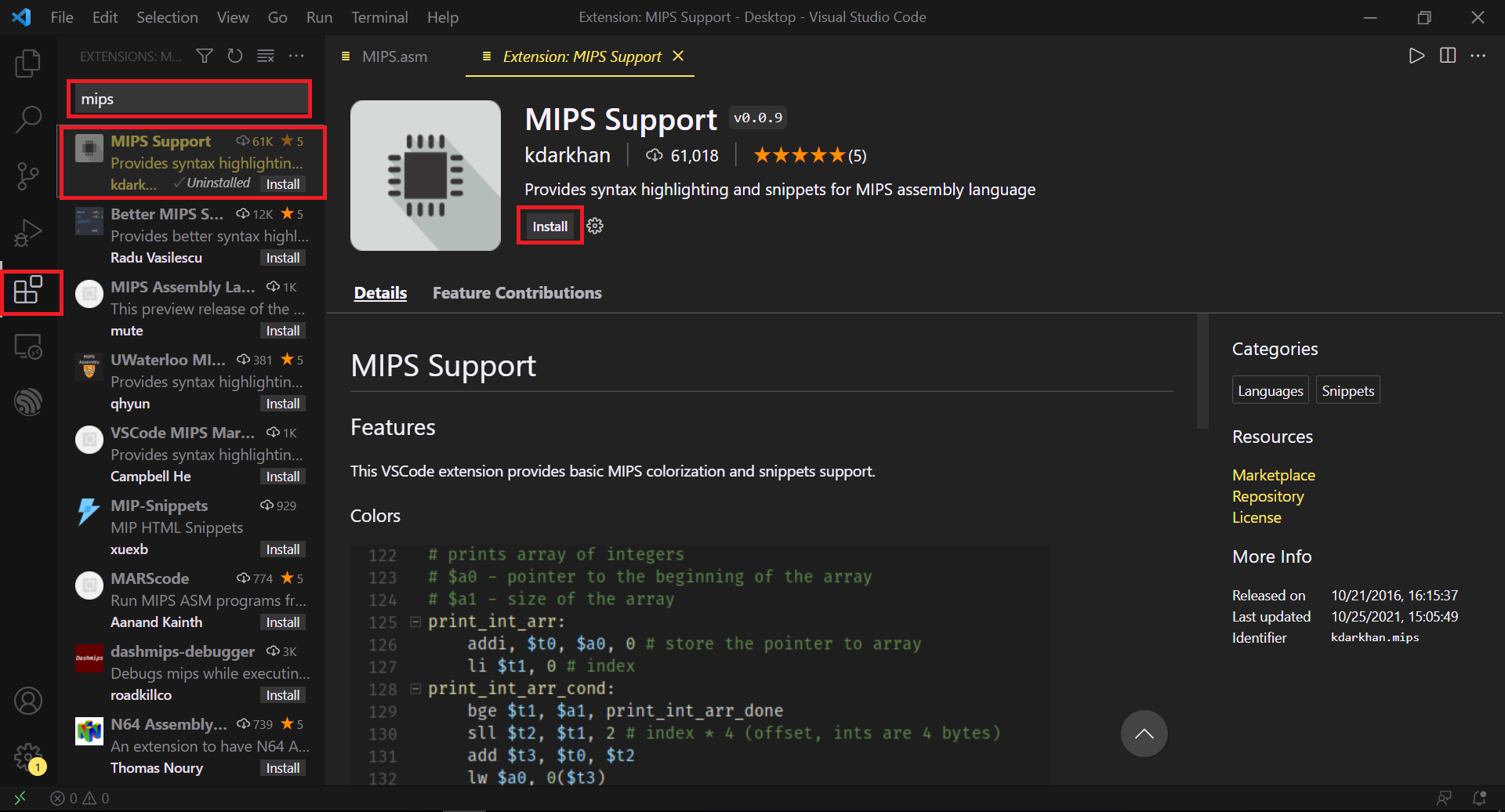
Similarly, you can install the VS code.

# VS Code Extensions:

Different extensions are used in VS Code to facilitate the programmer to write programs with ease and provide better readability in terms of text colors, auto-completion etc.

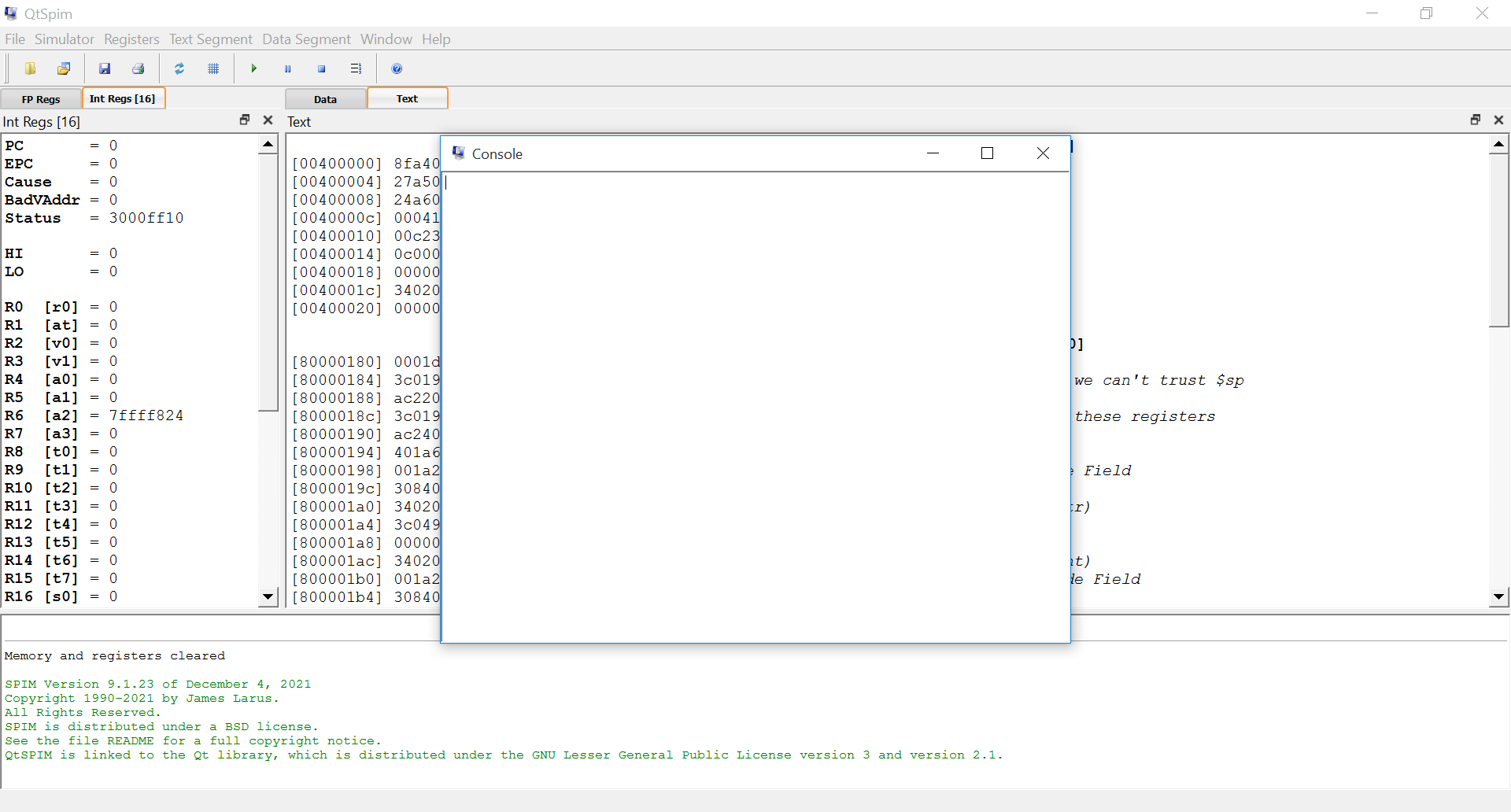
We need to install MIPS Support extension in VS Code for this purpose.

* Click on the “EXTENSIONS SYMBOL” on left side tool bar.
* Type MIPS SUPPORT in the search box.
* From the search results list, click the “MIPS SUPPORT”.
* Click on install button.
* The extension will be installed.



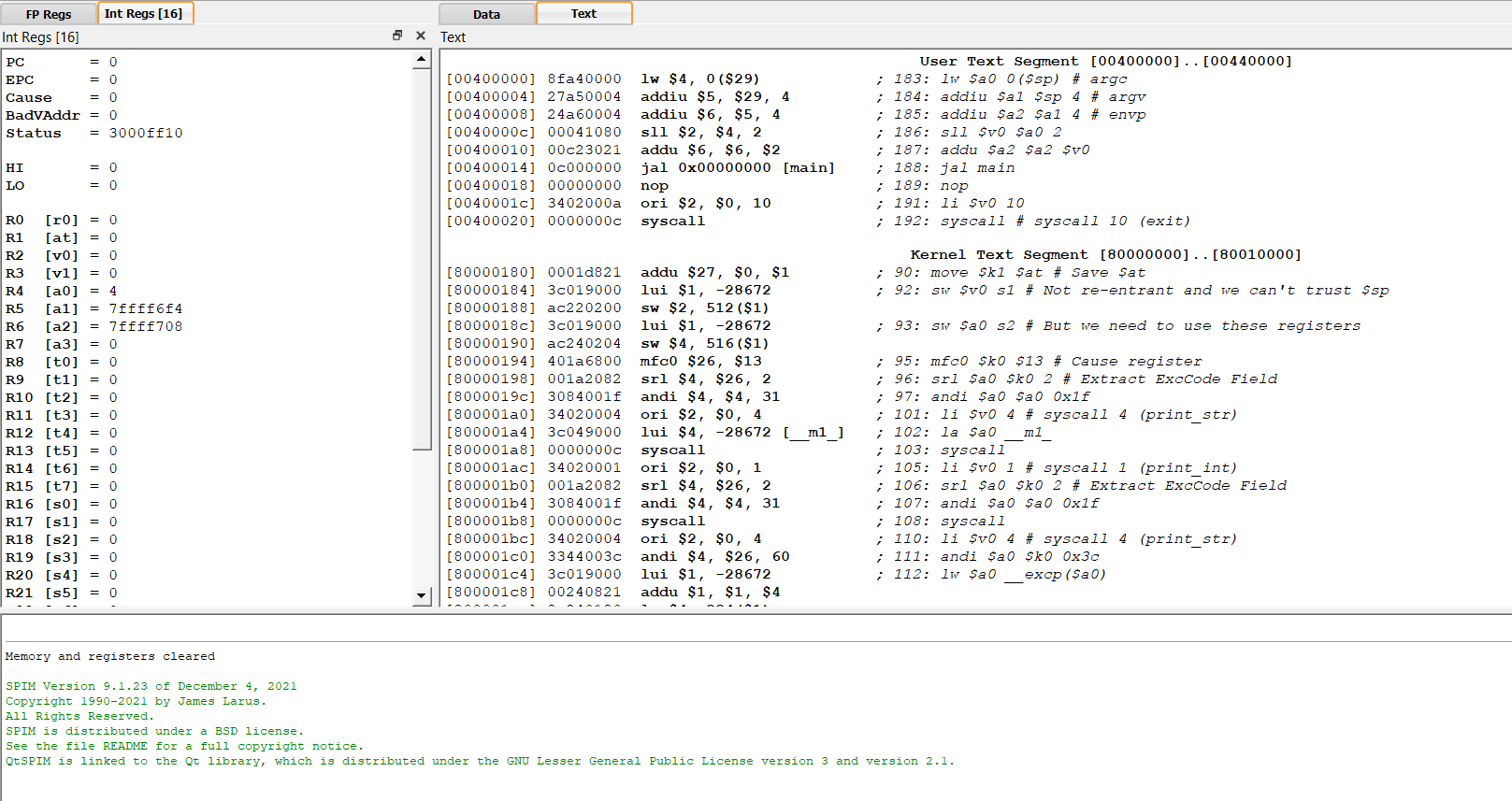
# Overview of QtSpim:

When QtSpims starts, two windows will open as shown in figure below. One window is for console which will display output information when you run your program. Another window is QtSpim main window.



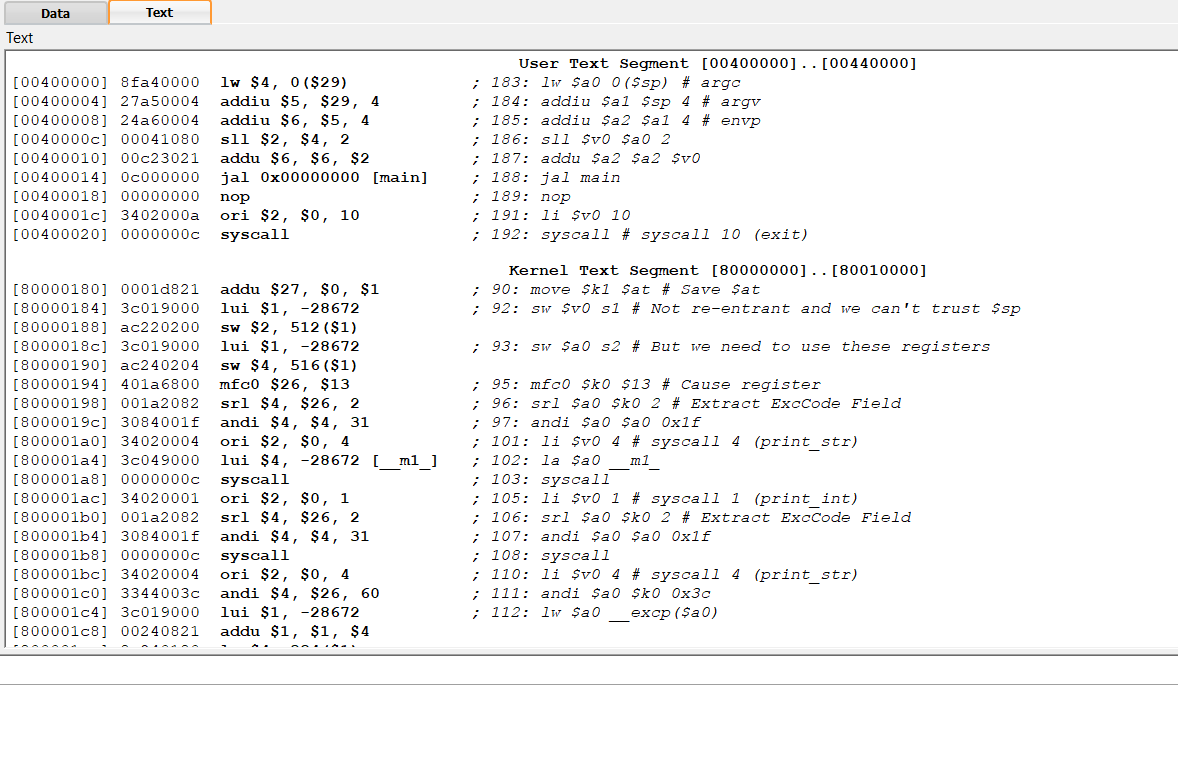
It is divided into three main parts as shown in figure below.

* The left pan which displays the registers and their values as the instruction gets executed. You can select either integer registers tab or floating point registers tab.
* The right big pan which displays the text segment (shows instructions) and data segment. You can select either text tab or data tab.
* The bottom pan which displays the QtSpim messages. If there is an error in your code or your code gets run successfully, it will show corresponding messages in this pan.



The text segment displays the following information (as shown in figure below):

* Memory Addresses of the instructions.
* OpCode of the instructions.
* Bare-Instructions and
* Pseudo-Instructions



Opcode

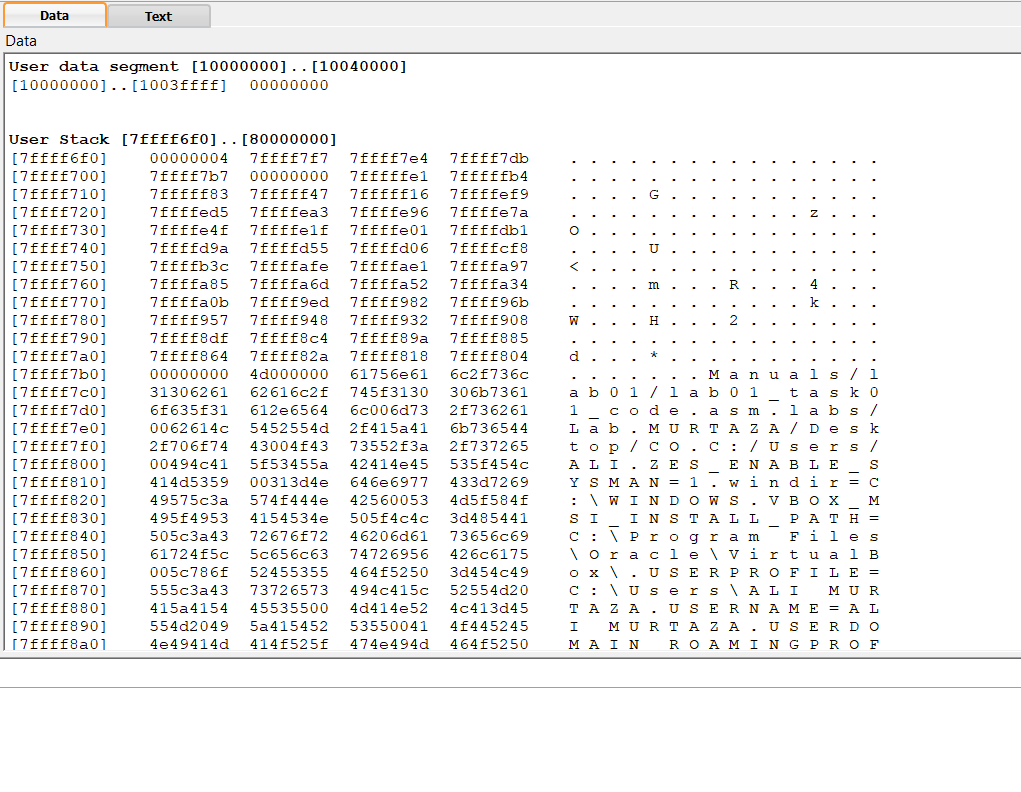
Pseudo-Instructions

Bare-Instructions

Addresses

The data segment displays the following information (as shown in figure below);

* Memory Addresses of the data
* Hex Representation of the data
* ASCII Representation of the data



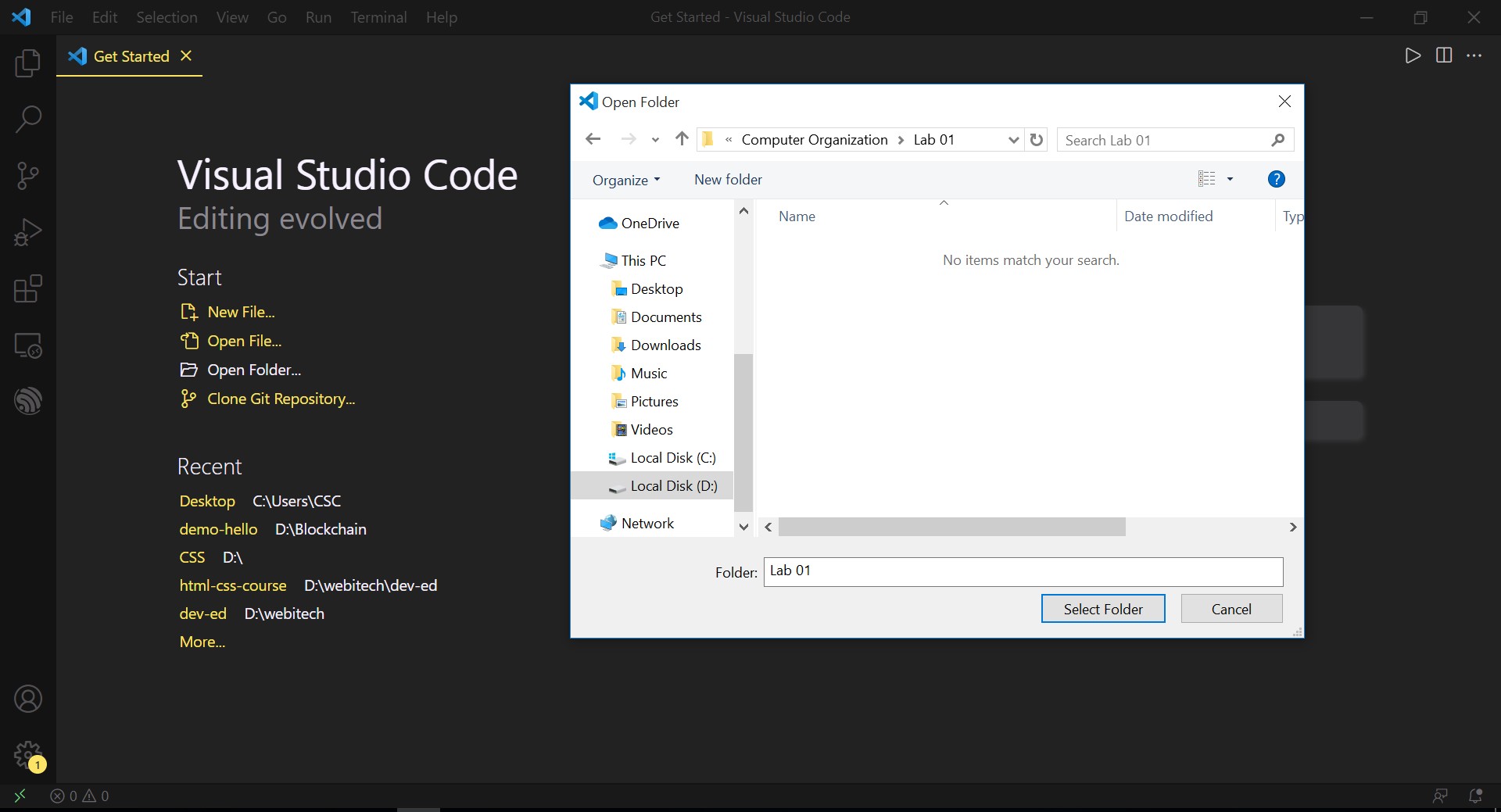
Addresses

Data (Hex Representation)

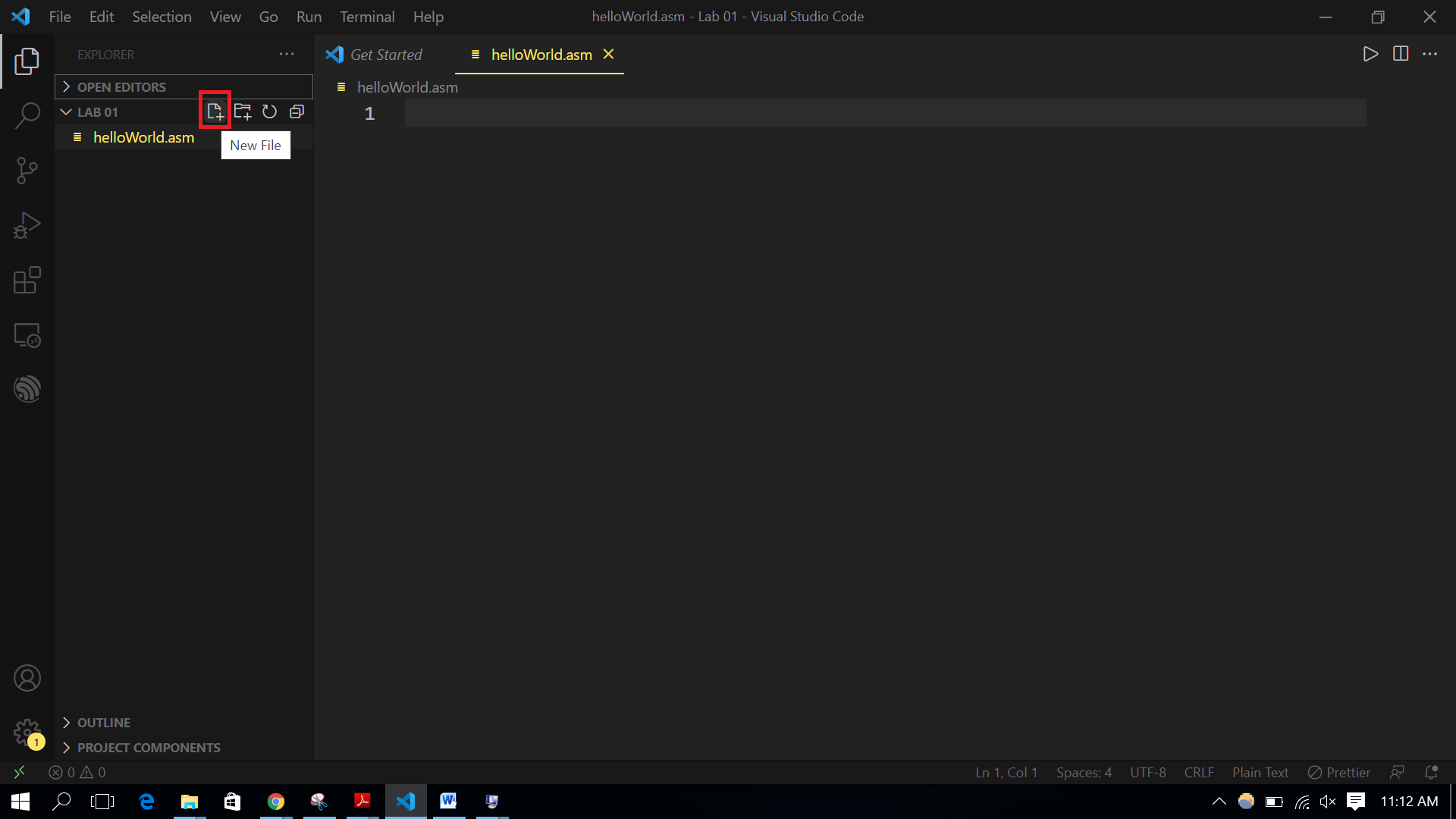
Data (ASCII Representation)

**Hello World Program:**

* Open the VS Code text editor and click on the “OPEN FOLDER”.
* In the pop up window, create a new folder at a suitable location. And select that folder as shown in the figure below.



* Then click on the “NEW FILE ICON” on left side pan of VS Code as shown in figure below. Type the name of the file for example “HelloWorld.asm”.
* It is better to use “**asm**” file extension.
* Double click on that file to open it.



* Type the following code.
* You can also copy it and paste it to the VS Code.

.data

message: .asciiz "Hello World!!\n"

.text

.globl main

.ent main

main:

li $v0, 4

la $a0, message

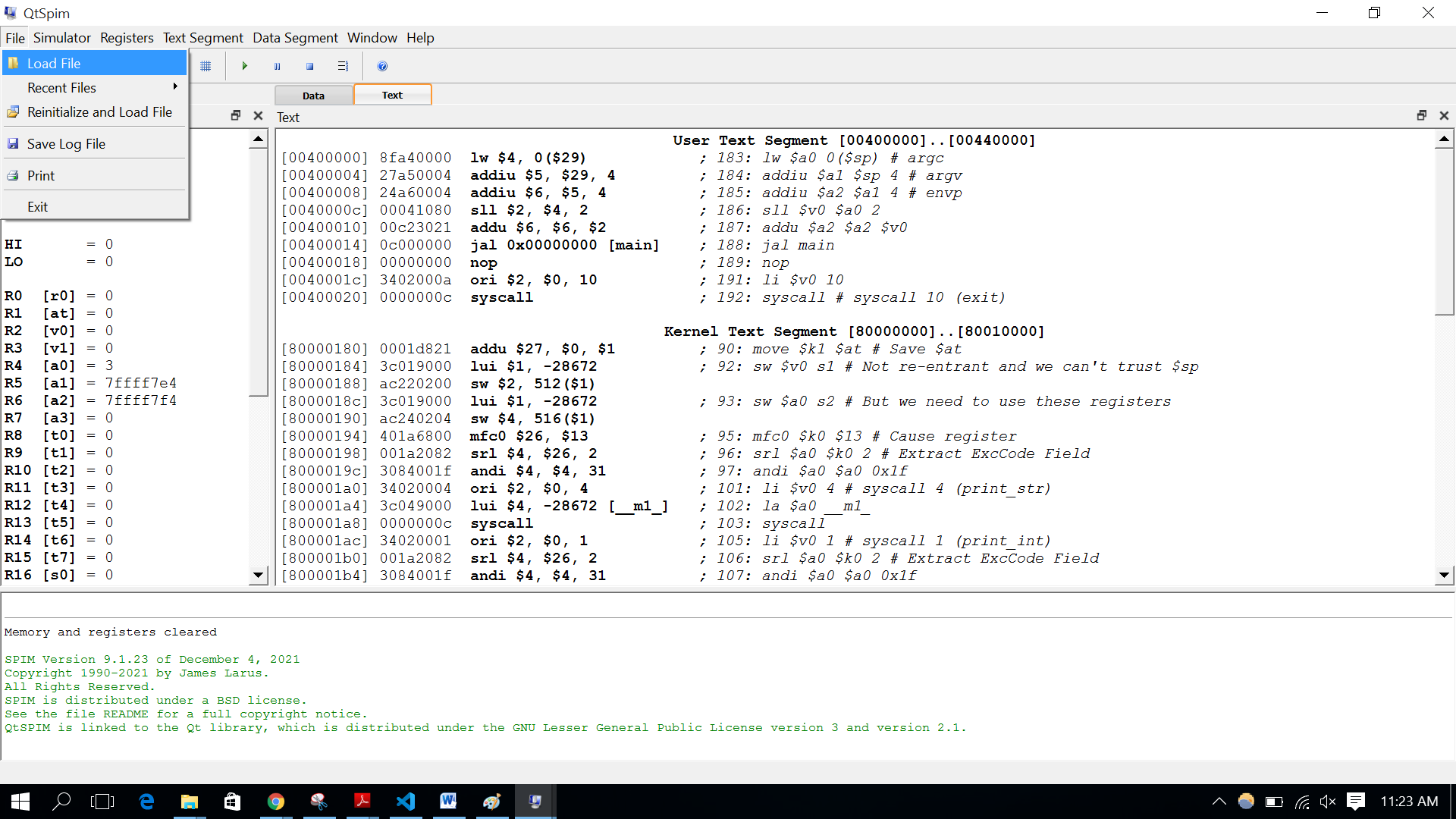
syscall

li $v0, 10

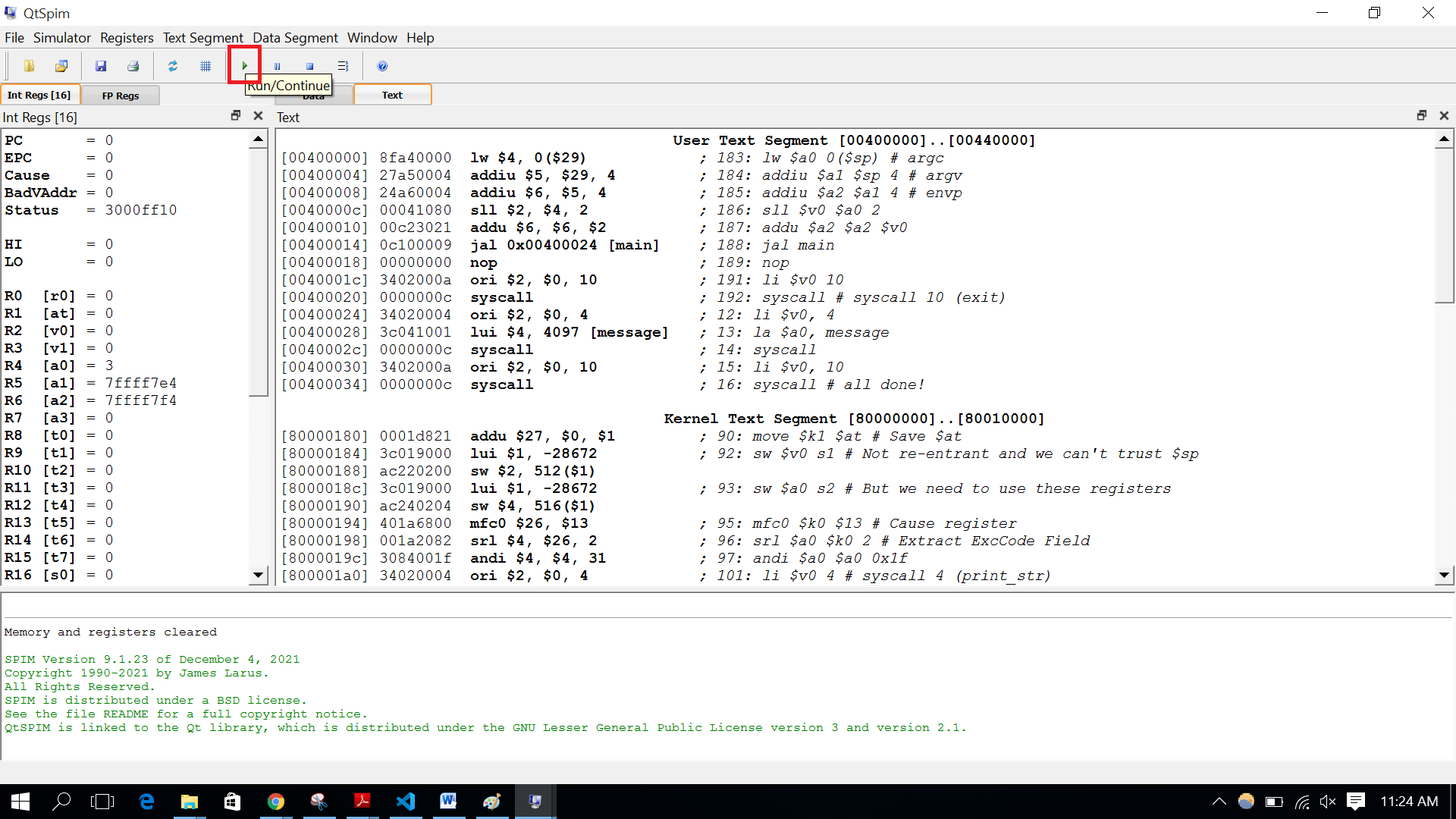
syscall # all done!

.end main

* Now save the file.
* Open the QtSpim and go to “file -> load file” and select that file which you have created as shown in the figure below.



* Run the code by clicking the run button as shown in the figure below.



* “Hello World!!” will be shown in the console window as shown below.

`

# Overview of Toolbar:

Each icon of the toolbar is explained below from left to the right:



* The first icon is used for loading the assembly program file, which needs to be executed by the QtSpim simulator.
* The second icon is used for reinitializing the simulator and loading the assembly program file.
* The third icon is used for saving the registers status (values), data segment, text segment and console output to the log file.
* The fourth icon is used to print the registers status (values), data segment, text segment and console.
* The fifth icon is used to clear the registers value.
* The sixth icon is used to reinitialize the simulator, which means you need to load the assembly program file again.
* The seventh icon (run/continue) is used to run/execute the whole program at once or at one click.
* The eighth icon (pause) is used to pause or halt the program execution. And you can continue the program execution by clicking the seventh icon (run/continue).
* The ninth icon (stop) is used to stop the program execution.
* The tenth icon (single step) is used to execute the code one line at one click. This way you can observe how the values of registers are changing by each instruction.

# Task 01:

You are provided with an assembly program file named “task01\_code.asm”. Load this file to the QtSpim simulator and execute it. And see the results on console window.

# Task 02:

Save the registers value and console output to the log file by using “save log file” icon.

# Task 03:

Now clear the registers value by clicking the “clear registers” icon. Now execute the same program again by clicking the “single step” icon and observe the program execution flow and how the values of registers are changing. Explain it in your words.

**Grading Scheme:**

|  |  |  |
| --- | --- | --- |
| Task# | Marks | Submission |
| Task1 | 5 | After running the code show the output to TA. |
| Task2 | 5 | Show the saved log file to TA and upload it on dropbox. |
| Task3 | 10 | Upload the explanation file on dropbox before leaving the lab. |