

PC SPIM Tutorial

SPIM is a self-contained simulator that will run MIPS R2000/R3000 assembly language programs. It reads and immediately executes MIPS assembly code. SPIM provides a simple debugger and minimal set of operating system services. Below are the steps to get a copy of SPIM:

1. Download the file <http://www.cs.wisc.edu/~larus/SPIM/pcspim.zip> and save it on your machine.
2. Unzip the file.
3. Run the *setup.exe* program.

To get more information about SPIM please visit: <http://www.cs.wisc.edu/~larus/spim.html>

Writing a program

Use Notepad/Wordpad (or any **text** editor) to write your MIPS assembly program. (PCSPIM requires one blank line after your last line of code -*this is no longer required for the new version*). So, ensure that you press the carriage return key at least once at the end of your program. Save the program with a ".s" or ".asm" extension. For example, *my_program1.s* or *my_program1.asm*.

Using the simulator

Double click on the PCSpim icon on your desktop. When the simulator is started, you should have four windows: *Registers*, *Text Segment*, *Data Segment*, and *Messages*. If you don't have them, click on the Window menu and select Tile.

Adjust simulation settings

Please change the simulation setting. Click on *Simulator* and choose *Setting*. A setting window will appear; please select *save window positions* (from *Display* setting), and *Allow pseudo instructions* and *Mapped I/O* (from *Execution* setting). Then, click OK.

Loading a program (open a file)

Select the Open file icon from the toolbar.

Alternatively, you can select from the menu bar: File→Open.

A file open dialog box will appear for you to select the appropriate assembly file. Select the appropriate assembly file and click on the button labeled *Open* in the dialog box. If simulator settings are not correct for the file, and it fails to load, PCSpim will provide you an opportunity to change simulator settings and automatically reload the file.

If you change your mind, click on the button labeled *Cancel*, and PCSpim removes the dialog box. When you load an assembly file, PCSpim removes dialog box, then loads your program and redraws the screen to display its instructions and data. If you have not done so, change the view of the four display windows to a tiled format by selecting from the menu bar: Windows→Tile. You should be able to see the program in the Text segment window display.

Each instruction in the Text segment window display is shown on a line that looks like:

[0x00400000]	0x8fa40000	lw \$4, 0 (\$29);	89: lw \$a0, 0(\$sp)
↑	↑	↑	↑
memory address (hex)	instruction (hex)	instruction (mnemonic)	Instruction from assembly file

The first number on the line, in square brackets, is the hexadecimal memory address of the instruction. The second number is the instruction's numerical encoding, again displayed as a hexadecimal number. The third item is the instruction's mnemonic description. Everything following the semicolon is the actual line from your assembly file that produced the instruction. The number 89 is the line number in that file.

Sometimes nothing is on the line after the semicolon. This means that the instruction was produced by SPIM as part of translating a pseudoinstruction.

Running a program

Click on the *Go* button in the toolbar.

Alternatively, you can select Simulator→Go from the menu bar.

Your program will begin execution. If you want to stop the execution of your program, select Simulator→Break from the menu bar. Alternatively, you can type Control-C when PCSpim application window is in focus. A dialog box will appear and ask if you want to continue execution. Select No to break the execution. Before doing anything, you can look at memory and registers contents in the Register display window to find out what your program was doing. When you understand what happened, you can either continue the program by selecting Simulator→ Continue or stop your program by selecting Simulator→ Break from the menu bar.

You may need to set PC (program counter) initial value. To do this select Simulator → set value. A dialog box will appear for you to select the appropriate register and value.

Address or register:	PC
Value:	0x00400000

If your program reads or writes from the terminal, PCSpim pops up another window called the console. All characters that your program writes appear on the console, and everything that you type as input to your program should be typed in this window.

Debugging a program (stepping & breakpoints)

SPIM has two features that help debug your program. The first is **single-stepping** which allows you to run your program an instruction at a time. Select Simulator→Single_Step to execute only one instruction. Alternatively, you can press the **F10** function key to single step. Each time you step through a program, PCSpim will execute the next instruction in your program, update the display, and return control to you. You can also choose the number of instructions in your program to step by selecting Simulator→ Multiple_Step instead of single stepping through your program. A dialog box will appear and ask you the number of instructions to step.

If your program runs for a long time before the bug arises, a better alternative is to use a **breakpoint**, which tells PCSpim to stop your program immediately before it executes a particular instruction. Select Simulator→Breakpoints from the menu bar. The PCSpim program pops up a dialog box window with two boxes. The top box is for you to enter breakpoint address and the second box is a list of active breakpoints. Type in the first box the address of the instruction at which you want to stop. Or, if the instruction has a global label, you can just type the name of the label. Labeled breakpoints are a particularly convenient way to stop at the first instruction of a procedure. To actually set the breakpoint, and click on the button labeled Add. When you are done adding breakpoints, click on the button labeled Close. You can then run your program. When the simulator is about to execute the breakpointed instruction, PCSpim pops up a dialog box with the instruction's address and asks if you want to continue the execution. The Yes button continues running your program and the No button stops your program. If you want to delete a breakpoint, you can select Simulator→ Breakpoints from the menu bar, click on the address in the dialog box, and click on the button labeled Remove.

Fixing bugs in a program

Single-stepping and setting breakpoints will probably help you find a bug in your program quickly. To fix a bug, go back to the editor that you used to create your program and change your source file. After you have made the changes to your source file, simply reload it into PCSpim for Windows by choosing Simulator→ Reload<filename> from the menu bar. This causes PCSpim to clear its memory and registers and return the processor to the state it was in when PCSpim first started. Once the simulator has reinitialized itself, it will reload your recently modified file.