

---

## SPIM simulator

CSE 410, Spring 2007  
Computer Systems

<http://www.cs.washington.edu/410>

4/3/2007

cse410-05-spim © 2006-07 Perkins, DW Johnson and University of Washington

1

---

## Reading and References

- See the Resources section on the SPIM web page for documentation on SPIM
  - » <http://www.cs.wisc.edu/~larus/spim.html>
- In particular, *Appendix A* (from the textbook) and *Getting Started with PCSpim* are useful reading from that site

4/3/2007

cse410-05-spim © 2006-07 Perkins, DW Johnson and University of Washington

2

---

## SPIM simulator

- SPIM lets you write MIPS assembly language code and run it on a PC
- PCSpim is installed on the machines in the Math Sciences Computing Center
- You can download versions for Windows and all varieties of \*nix (including MacOS X) from the web site
  - » <http://www.cs.wisc.edu/~larus/spim.html>

4/3/2007

cse410-05-spim © 2006-07 Perkins, DW Johnson and University of Washington

3

---

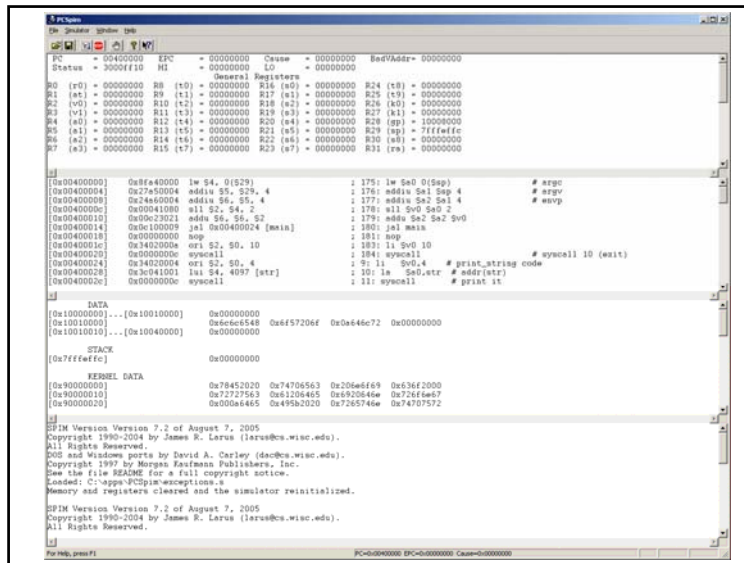
## Spim display

- Register panel
  - » register names and numbers
- Text segment panel
  - » note jump and link to “main” at [0x00400014]
  - » your code defines the label “main”
- Data and Stack segment panel
- Message panel

4/3/2007

cse410-05-spim © 2006-07 Perkins, DW Johnson and University of Washington

4



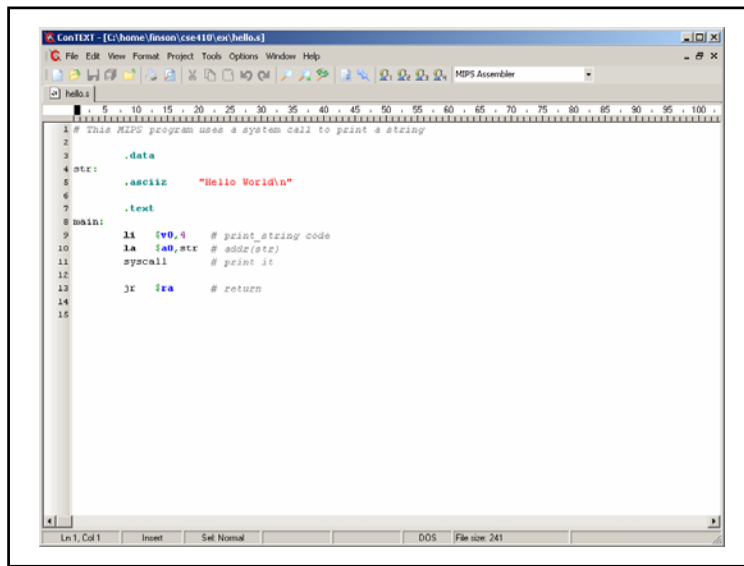
## Context editor

- You can use any text editor you like to write the source code
  - » see links on class software page
- Context editor provided in MSCC
  - » it has a highlighter for MIPS assembly language
  - » it doesn't try to be a word processor
- jEdit also provides a MIPS highlighter
- emacs can do anything including asm – but has a huge learning curve

4/3/2007

cse410-05-spm © 2006-07 Perkins, DW Johnson and University of Washington

6



## hello.s

```

.data
str:
.asciiz "Hello World\n"

.text
main:
li $v0, 4 # print_string code
la $a0, str # addr(str)
syscall # print it

jr $ra # return

```

4/3/2007

cse410-05-spm © 2006-07 Perkins, DW Johnson and University of Washington

8