

CSC 553: Principles of Compilation

The MIPS R2000 instruction set and SPIM

The MIPS R2000 offers a RISC instruction set architecture (ISA) whose primary attraction, from the perspective of this class, is that it is small, clean, and simple; SPIM is an open-source software implementation of an emulator for this ISA. By contrast, using something like the Intel x86-64 ISA would involve significantly more time and effort because of its complexity.

1. Finding spim

On the CS department server **lectura**, the executable for SPIM is available at

```
/bin/spim
```

However, this software does not allow us to easily quantify the effects of optimizations. So for the purposes of this class we'll use a variant of spim that keeps track of the number of times various different kinds of instructions are executed. This variant is available at

```
/home/cs553/fall22/bin/spim      (executable)
/home/cs553/fall22/spim-stats    (source code)
```

Note: If you want to build your own copy of this spim variant, you should proceed as follows:

1. Copy over the source code directory mentioned above.
2. Install two open-source software packages, flex and bison, that are used by spim's front end. (Installation instructions for Ubuntu can be found [here](#).)
3. `cd spim-stats/spim`
4. Edit `Makefile`, in particular edit the value of the variable `PREFIX`
5. Type `'make spim'` -- this will create an executable file `spim` in that directory. If you want you can copy or move this executable to some other place that you prefer.

2. Using spim

To run a MIPS assembly code program that is in a file `foo`, execute the command

```
spim -f foo
```

To get instruction counts for a run, you will have to use the `spim-stats` variant mentioned above. With this variant, you can get execution counts using the command:

```
spim -keepstats -f foo
```

A user manual for SPIM is available at <http://www2.cs.arizona.edu/classes/cs553/spring21/DOCS/spim.pdf>. This user manual contains a description of all the instructions available in the MIPS R2000 ISA.

3. Debugging in SPIM

Bugs in your compiler can result in the generation of incorrect SPIM asm code. In such cases, in order to identify and fix your compiler bug, you first have to identify the problem in the generated SPIM code.

The SPIM user manual, **Section 1.2.1, "Terminal Interface"**, describes several commands that can be used for debugging purposes. In particular, you can set and remove breakpoints, single step through the execution, and print the contents of registers and memory.