

## Getting Started with spim

The simple version of SPIM is called `spim` on Windows and UNIX. It can be run from any console or by another program. Although `spim` may be more difficult to learn, it operates just like `PCSpim` and `xspim` and provides the same functionality.

The `spim` terminal interface provides the following commands:

<code>exit</code>	Exit the simulator.
<code>read "file"</code>	Read <i>file</i> of assembly language into SPIM. If the file has already been read into SPIM, the system must be cleared (see <code>reinitialize</code> , below) or global labels will be multiply defined.
<code>load "file"</code>	Synonym for <code>read</code> .
<code>execute "a.out"</code>	Read the MIPS executable file <i>a.out</i> into SPIM. This command is only available when SPIM runs on a system containing a MIPS processor.
<code>run &lt;addr&gt;</code>	Start running a program. If the optional address <i>addr</i> is provided, the program starts at that address. Otherwise, the program starts at the global label <code>__start</code> , which is usually the default start-up code that calls the routine at the global label <code>main</code> .
<code>step &lt;N&gt;</code>	Step the program for <i>N</i> (default: 1) instructions. Print instructions as they execute.
<code>continue</code>	Continue program execution without stepping.
<code>print \$N</code>	Print register <i>N</i> .
<code>print \$fN</code>	Print floating-point register <i>N</i> .
<code>print_all_regs</code>	Print all registers.
<code>print_all_regs hex</code>	Print all registers in hexadecimal.
<code>print addr</code>	Print the contents of memory at address <i>addr</i> .
<code>print_sym</code>	Print the names and addresses of the global labels known to SPIM. Labels are local by default and become global only when declared in a <code>.globl</code> assembler directive (see "Assembler Syntax" section on page A-47).
<code>reinitialize</code>	Clear the memory and registers.

<code>breakpoint addr</code>	Set a breakpoint at address <i>addr</i> . <i>addr</i> can be either a memory address or symbolic label.
<code>delete addr</code>	Delete all breakpoints at address <i>addr</i> .
<code>list</code>	List all breakpoints.
<code>.</code>	Rest of line is an assembly instruction that is stored in memory.
<code>&lt;n1&gt;</code>	A newline reexecutes previous command.
<code>?</code>	Print a help message.

Most commands can be abbreviated to their unique prefix (e.g., `ex`, `re`, `l`, `ru`, `s`, `p`). More dangerous commands, such as `reinitialize`, require a longer prefix.