

NAME

xspim - A MIPS R2000/R3000 Simulator

SYNTAX

```
xspim [-asm/-bare -trap/-notrap -quiet/-noquiet
-mapped_io/-nomapped_io -file file -execute file
-stext size -sdata size -sstack size -sktext size -skdata
size -ldata size -lstack size -lkdata size
-hexgpr/-nohexgpr -hexfpr/-nohexfpr]
```

DESCRIPTION

SPIM S20 is a simulator that runs programs for the MIPS R2000/R3000 RISC computers. (For a description of the real machines, see Gerry Kane and Joe Heinrich, *MIPS RISC Architecture*, Prentice Hall, 1992.) SPIM can read and immediately execute files containing assembly language or MIPS executable files. SPIM is a self-contained system for running these programs and contains a debugger and interface to a few operating system services.

SPIM comes in two versions. The plain version is called *spim*. It runs on any type of terminal. It operates like most programs of this type: you type a line of text, hit the *return* key, and *spim* executes your command. The fancier version of SPIM is called *xspim*. It uses the X-window system, so you must have a bit-mapped display to run it. *xspim*, however, is a much easier program to learn and use because its commands are always visible on the screen and because it continually displays the machine's registers.

OPTIONS

xspim has many options:

```
-asm      Simulate the virtual MIPS machine provided by the
          assembler. This is the default.

-bare     Simulate a bare MIPS machine without pseudo-
          instructions or the additional addressing modes
          provided by the assembler. Implies -quiet.

-trap     Load the standard exception handler and startup
          code. This is the default.

-notrap   Do not load the standard exception handler and
          startup code. This exception handler handles
          exceptions. When an exception occurs, SPIM jumps
          to location 0x80000080, which must contain code to
          service the exception. In addition, this file
          contains startup code that invokes the routine
          main. Without the startup routine, SPIM begins
          execution at the instruction labeled __start.

-quiet    Print a message when an exception occurs. This is
          the default.

-noquiet  Do not print a message at exceptions.

-mapped_io Enable the memory-mapped IO facility. Programs
          that use SPIM syscalls to read from the terminal
          cannot also use memory-mapped IO.

-nomapped_io Disable the memory-mapped IO facility.

-file     Load and execute the assembly code in the file.
```

```
-execute  Load and execute the MIPS executable (a.out) file.
          Only works on systems using a MIPS processors.
```

```
-stext size -sdata size -sstack size -sktext size -
skdata size
Sets the initial size of memory segment seg to be
size bytes. The memory segments are named: text,
data, stack, ktext, and kdata. The text segment
contains instructions from a program. The data
segment holds the program's data. The stack seg-
ment holds its runtime stack. In addition to run-
ning a program, SPIM also executes system code
that handles interrupts and exceptions. This code
resides in a separate part of the address space
called the kernel. The ktext segment holds this
code's instructions and kdata holds its data.
There is no kstack segment since the system code
uses the same stack as the program. For example,
the pair of arguments -sdata 2000000 starts the
user data segment at 2,000,000 bytes.
```

```
-ldata size -lstack size -lkdata size
Sets the limit on how large memory segment seg can
grow to be size bytes. The memory segments that
can grow are data, stack, and kdata.
```

```
-hexgpr   Display the general purpose registers (GPRs) in
          hexadecimal.
```

```
-nohexgpr Display the general purpose registers (GPRs) in
          decimal.
```

```
-hexfpr   Display the floating-point registers (FPRs) in hex-
          adecimal.
```

```
-nohexfpr Display the floating-point registers (FPRs) as
          floating-point values
```

BUGS

Instruction opcodes cannot be used as labels.

SEE ALSO

```
spim(1)
James R. Larus, ``SPIM S20: A MIPS R2000 Simulator,''
included with SPIM distribution.
```

AUTHOR

James R. Larus (larus@cs.wisc.edu), Computer Sciences
Department, University of Wisconsin-Madison.

