

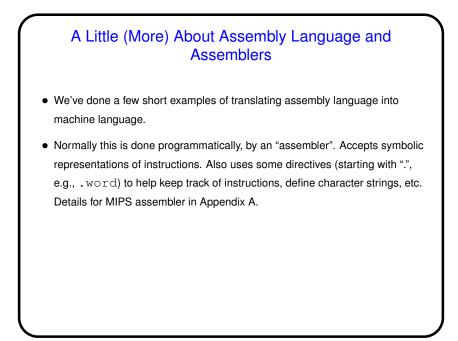
Pseudo-Direct Addressing Address is formed by combining address in instruction (26 bits) and upper bits of program counter. (Actually, address is address in instruction times 4, or'd with upper bits of program counter.) Slide 6 Example is unconditional branch (j). Does this limit what we can do with j? If so, will that be a problem? Can we work around it?

Example

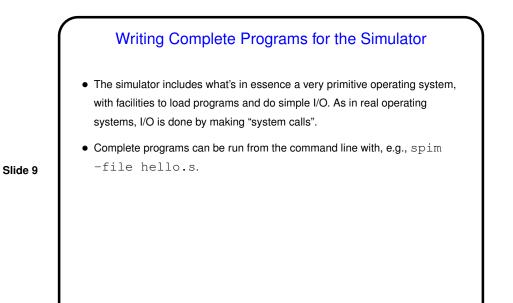
• As an example, try translating the following C first into MIPS assembly language and into machine language. (Assume registers \$s0 and \$s1 are being used for a and b.)

```
if (a < b)
a = a + 1;
else
b = b - 1;
```

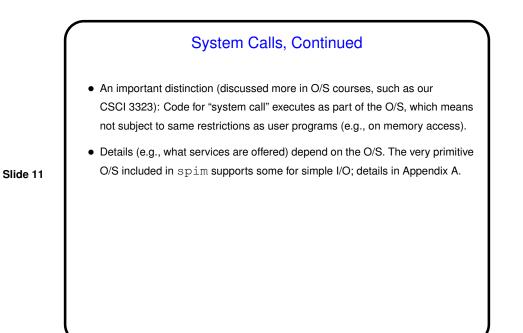
Slide 7



Slide 8



System calls are how user programs request service from the operating system — not just in MIPS, but in general. In MIPS the instruction is syscall; other architectures have something analogous.
System calls similar to procedure calls in some ways — need to communicate to O/S which service you want (e.g., write some text to "standard output") and possibly parameters (e.g., the text to write). As with procedure calls, we do this by putting values in particular registers, but then rather than jal we use syscall.



Complete Programs — Examples
We can now write some simple but complete programs for the simulator(!).
(Examples on "sample programs" page.)

	Minute Essay
	 What does the following code do? i.e., what is in registers \$s0 and \$s1 after it executes?
	add \$s0, \$zero, \$zero
	addi \$s1, \$zero, 1
de 13	addi \$s2, \$zero, 4
	label:
	addi \$s0, \$s0, 1
	add \$s1, \$s1, \$s1
	bne \$s0, \$s2, label

Slid

