

## Arrays of Text Strings and Command-Line Arguments If you can have arrays of int and char and so forth — can you have arrays of text strings? Sure! They look like two-dimensional arrays of char, or like arrays of char \*. Further, this is how C programs get input "from the command line" (e.g., when you write gcc myprogram.c, gcc somehow gets myprogram.c, right?): main can also be defined as int main(int argc, char \* argv[]) { .... } where argc is the number of arguments, plus one, and argv is an array of strings containing the arguments. Example — let's write a simple "echo" program.



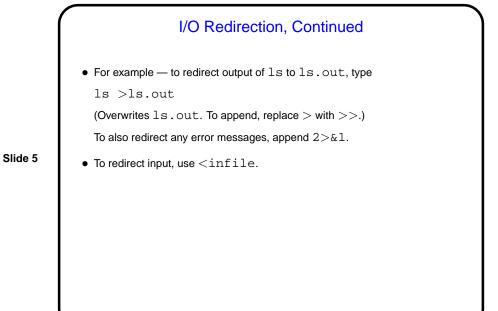
• In programming classes I talk about "reading from standard input" rather than "reading from the keyboard", and "writing to standard output" (or "writing to standard error") rather than "writing to the screen".

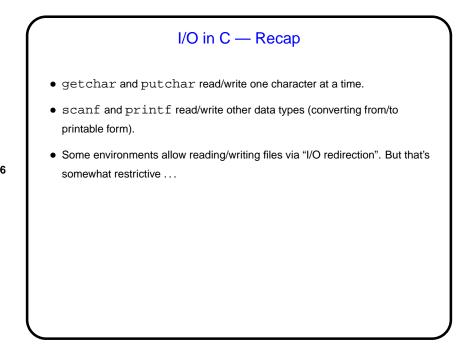
(In Java terms — System.in, System.out, and System.err. C has similar concepts but calls them stdin, stdout, and stderr.)

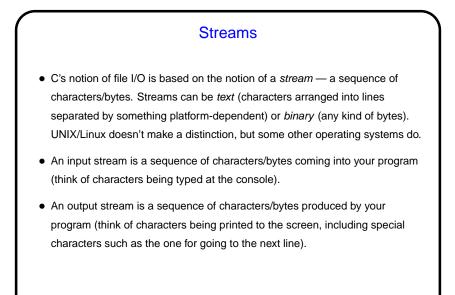
• What's the difference?

I/O Redirection, Continued
stdin (standard input) can come from keyboard, file, or from another program or shell script.
stdout and stderr (standard output, error) can go to terminal or file (overwrite or append), separately or together.

Slide 4







Streams in C
In C, streams are represented by the type FILE \* - i.e., a pointer to a FILE, which is something defined in stdio.h.
A few streams are predefined - stdin for standard input, stdout for standard output, stderr) for standard error (also output, but distinct from stdout so you can separate normal output from error messages if you want to).
To create other streams ...

