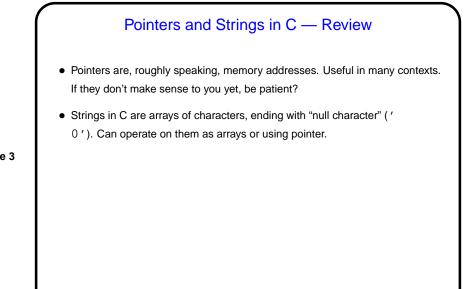
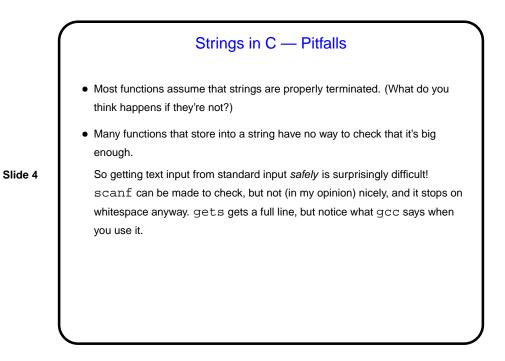


Minute Essay From Last Lecture
One student asked which of the two "string length" functions should be used. Best answer is "neither one" — there's a library function strlen. (But the ones we wrote are meant as examples of working with strings.)
But the general question of accessing arrays by index or via pointers ... There was a time when the pointer version was likely to be more efficient. Probably no longer true; "good" compiler will likely generate the same code for both.



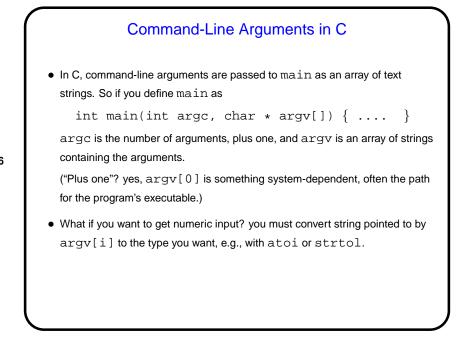


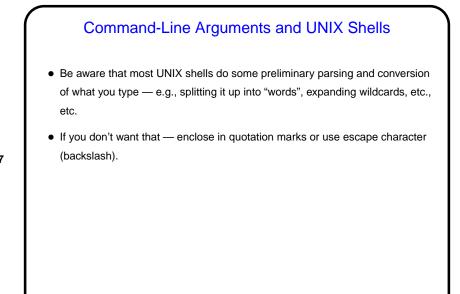


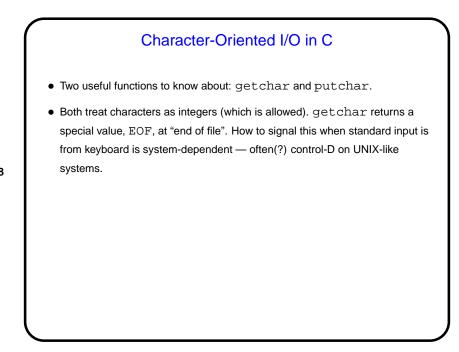
Now that we know about arrays, pointers, and text strings, we can talk about command-line arguments. What are they? text that comes after the name of the program on the command line (e.g., when you write gcc -Wall myprogram.c, there are are two command-line arguments), possibly modified by the shell (e.g., for filename wildcards).

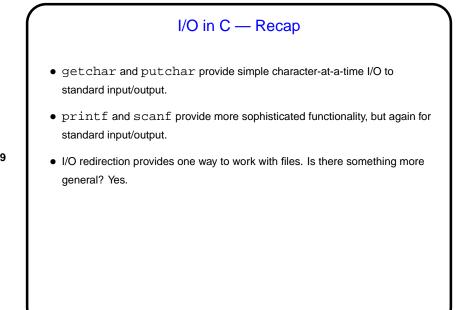
Slide 5

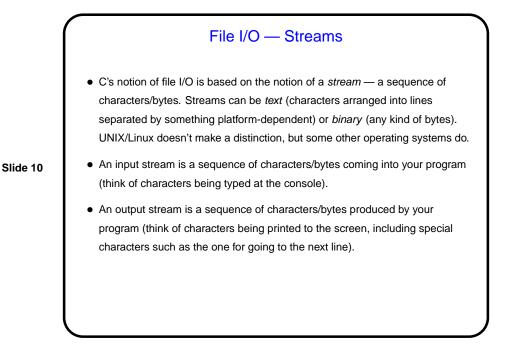
 Most programming languages provide a way to access this text, often via some sort of argument to the main function/method.











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 ...

Creating Streams in C
To create a stream connected with a file — fopen.
Parameters, from its man page:

First parameter is the name of the file, as a C string.
Second parameter is how we want to access the file – read or write, overwrite or append — plus a b for binary files, also a string.
Return value is a FILE * — a somewhat mysterious thing, but one we can pass to other functions. If NULL, the open did not succeed. (Can you think of reasons this might happen?)

