

# CS-701 Lecture 21

April 24, 2001

## Administrivia

- Assignment 4 due by midnight tonight.
  - 1% per day late; cutoff will be May 1.
  - Fix bugs from Assignment 3 that caused memory dumps or erroneous execution.
  - See Blackboard for requirements about implementing the *which* command.
  - No need for a README file, unless there is something unusual about your project that you want me to know about, such as extra features I might otherwise miss.

## Global Data and Functions

- Functions and data may be declared external (*extern*) in any number of source files, so it's appropriate to put the *externs* in a header file.
  - Doing this assures consistency across different .cc files.
- The data or function must be defined in exactly one .cc file. (When would an error about this be detected?)
- Local variables cannot be accessed from other functions, to say nothing of other source files.
  - Don't define the dispatch table inside *main()*.
- Static variables can be accessed only from the source file in which they are defined.
- Declarations may refer to other types and variables that are already declared.

## Redirect I/O for Builtins

(Not Required)

- Use *dup()* to create a copy of fd to be redirected.
- Close the fd to be redirected.
- Open the file for redirection.
- Read/write.
- Close the redirected fd and use *dup()* to get back the original value.
  - Better: Use *dup2()* to combine these two operations.
- Stevens, pages 61-63.

## Reading a Command Line

- Goal is to get an array of strings representing command-line tokens.
  - Allow quotes, redirection symbols, pipes, semicolons, ampersands (, ...?) as tokens.
  - Allow user to edit the command line.
  - Support arrow keys.
- Low-level line reader
  - Different from *fgets()* in that the program examines each character as it is typed.
    - Advantage: Support for arrow keys
    - Disadvantage: Have to write the line editing functions
- High-level line processor
  - Takes the line returned by the low-level reader and tokenizes it.
    - Has to process the line a character at a time