CS-701 Lecture 4

February 8, 2001

Dr. Vickery

Some Links

- David Korn on Slashdot
- Typescript of Assignment 1
- Coding Guidelines
- printargs.c
- Typescript Showing RCS

Building an Executable File

- *Compiler driver* manages a pipeline of commands.
  - Preprocess (*cpp*)
  - Compile to to assembly language (*cc1* or *cc1plus*)
  - Assemble to an object module (*as*)
  - Link object modules and libraries (*ld*)
- File suffixes (.c and .o in particular) tell the compiler driver where to start in the pipeline.
- Some options tell where to stop the pipeline.
  - -E (stop after *cpp*); -S (stop after *c1*); -c (stop after *as*)
- Other important options are passed to commands in the pipeline.
  - -I, -D (passed to *cpp*)
  - -o, -l, -L (passed to *ld*)

The *man* Comand

- *man* <command, function, etc.>
- "The Manual" is divided into sections
  - 1 = Commands
  - 2 = System Calls
  - 3 = Library Functions
  - And others …
  - Example: *man 2 read*

The Make Utility

- Command: *make* <target>
- Built-in Rules
  - Based on file suffixes. It knows how to convert .c files into executable files, but doesn't know how to make .c files, because people do that using an editor.
- Environment Variables
  - You have to set the CC and CFLAGS environment variables to get *make* to generate the correct *gcc* command for you.
    - setenv CC gcc
    - setenv CFLAGS "-g –Wall"
- More complicated rules
  - Use a Makefile

tar and *script* Commands

- *tar* – "Tape Archiver" – See Assignment 1 sample typescript for example.
  - *tar cvf somefile.tar <files>*
  - *tar tvf somefile.tar*
  - *tar xvf somefile.tar*
- *script*
  - Captures console session in a file named typescript.
  - Use *exit* to, uh, exit.