

CSC295 2014S, Class 05: Compilation, Macros, and the C Preprocessor

Overview

- Preliminaries.
 - Admin.
 - Questions.
- Topics for today
 - A C program ecosystem.
 - Steps in compilation.
 - Including files.
 - *Detour: Why we have included files.*
 - Macros.
 - CPP conditionals.
 - *Detour: Standard header file format.*
 - Command-line flags.
 - Hack: Generic data types.

Preliminaries

Admin

- Apologies for not being able to be in class last week.
- Sorry for the confusion about last week and this. I think things are now resolved.
- This week's assignment: Find an interesting use of macros and/or includes in C code in the wild and document it.
- Happy Thunderhail!
- Cool CS activities:
 - Thursday extra: Technical interviews.
 - CS Table Friday: Skip lists (if I ever find the paper).

A C program ecosystem

- Header files
 - provide prototypes for functions
 - some written by the person who is working on this project
 - others are part of the standard installation
 - others come from library code that you've installed
- Source code

- Your utility/library code
- Proto-executables
- External things
 - The utility code (traditionally put in library files, .a or .so)
 - The tools that let you build your program (gcc, make, valgrind, gdb, lint, gprof)
- Unit tests!
- Documentation (at least a README) to explain the parts to someone else
 - Automatically generated from the code
- A Makefile that ties everything together
- An appropriate license so that people know what they might be able to do with your code. Also an appropriate disclaimer of fitness.
- Remember that there's a lot of stuff as you think about building programs.

Steps in compilation

- Start with source (.c)
- Preprocess - simple textual manipulation (cc -E)
 - Replace #include with the included file (plus notes)
 - Replace constant definitions with the defined value

define MAXARRAYSIZE 24

int values[MAXARRAYSIZE];

int values[24];

- Macro processing

define MAX(X,Y) (X > Y) ? X : Y

y = MAX(a,b);

y = (a > b) ? a : b;

The following is of questionable legality

(MAX(a,b)) = y; ((a > b) ? a : b) = y;

The following is dangerous

```
y = MAX(a++,b++)
```

- Conditional code `#ifdef NAME fprintf(stderr, "Hi mom!"); #endif`
- Output: C code, but longer
- Translation into a simpler language
 - Assembly (.s)
- Assemble to relocatable machine code
 - Machine code (.o)
- Link together into an executable
 - Executable

Suppose I have a call to `foo()` in `main.c`

- The compiler doesn't know where in the executable "foo" will reside.
- So it generates a "jump to `_foo`" instruction

The linker figures out where all the instructions go (and what instructions you need) and replaces the `_foo` with the address.

Including files

```
#include <stdio.h>
#include "foo.h"
```

The angle brackets say "It's an installed header file" The lack of angle brackets say "It's a relative path"

Where does it find the `stdio.h`?

- By default: `/usr/include` (or whatever your compiler is set to use)
- Probably some environment variable
- Probably some Make environment variable
- `-I`
 - on the command-line

Detour: Why we have included files

- See the example.

Macros

CPP conditionals

Detour: Standard header file format

Command-line flags

Hack: Generic data types

See `examples/generic`

Copyright (c) 2013-14 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.