

ECE 131 Programming Fundamentals

Dr. Daryl Lee

University of New Mexico



The C Preprocessor

- So what's this `#include <stdio.h>` all about?
- Before the compiler compiles, the preprocessor preprocesses.
- Every line that starts with `#` means something to the preprocessor
- Good choice of preprocessor usage makes a programmer more powerful.
- We'll return to `#include`; but first, now for something completely different.

#define

- After `#include`, the most commonly used preprocessor directive is `#define`. Its uses are:
 - Constant definition
 - `#define BUFFER_LENGTH 80`
 - `#define FILE_NAME "mydata.txt"`
 - `#define WIN`
 - Note: No "=", No ";"
 - No "value" actually needed
 - Macro definition
 - `#define MAX(x,y) ((x) > (y) ? (x) : (y))`

Textual Substitution

- `#define` works by textual substitution.
- When the preprocessor encounters a symbol that has been `#define`'d, it simply substitutes the text that the symbol was defined with.
- If the symbol had associated parameters, the parameters are also textually substituted.

Example One

- Example:

```
#define MAX_BUFFER 80
```

```
...
```

```
char line[MAX_BUFFER + 1];
```

The last line is “preprocessed” and the compiler sees

```
char line[80 + 1];
```


Example Two

- Example:

```
#define MAX(x,y) ((x) > (y) ? (x) : (y))
```

```
...
```

```
z = MAX(r + 3, 2 * s);
```

The last line is “preprocessed” and the compiler sees

```
z = ((r + 3) > (2 * s) ? (r + 3) : (2 * s));
```

#ifdef, #ifndef, #else, #endif

- Conditional compilation
 - Example

```
#ifdef WIN
    // Windows code goes here
#else
#ifdef MACOSX
    // Mac code goes here
#else
#ifdef LINUX
    // Linux code goes here
#endif // LINUX
#endif // MACOSX
#endif // WIN
```

#if, #elif, #endif

- Comparisons like `>`, `<`, `==`, `!=` allowed
- Example:

```
#define OS 1
...
#if OS == 1
...
#elif OS == 2
...
#endif
```
- Defining on compiler command line:

```
$ gcc -D OS=1 myprogram.c
```


#include

- Inserts contents of the specified file in its place
- “...” causes search to start locally
- <...> causes search to start in system-defined locations
- Recursive; i.e., included files may have #include directives in them.

Include guards

- Compiler does not allow redefining symbols.
- If a `#include` chain causes the same symbol to be redefined, an error occurs. How to prevent?

- Classic:

```
#ifndef FILENAME_H
#define FILENAME_H
...
#endif
```

- Modern, but unreliable:

```
#pragma once
```