The C Preprocessor

ITSC 2181: Introduction to Computer Systems
UNC Charlotte
College of Computing and Informatics

COLLEGE OF COMPUTING
AND INFORMATICS

Preprocessing

- Modifies the contents of the source code file **before** compiling begins
- The preprocessor is run automatically when you compile your program
  - Use the `gcc -E` option if you want to see **just** the results of the preprocessing step
- It is (mostly) simple **string substitution**

(see `constant.c` in *Code Samples and Demonstrations* in *Canvas*)

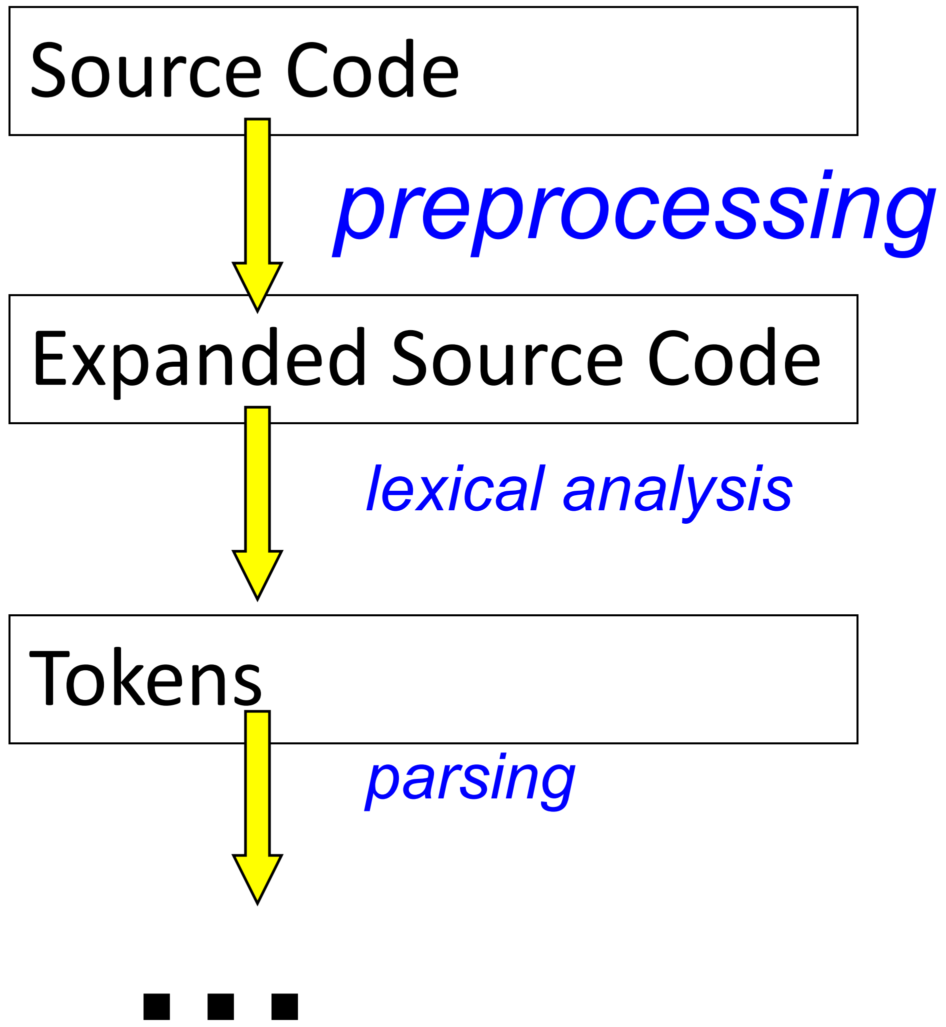
```
#define PI 3.1415926
double x = PI * d;
```

preprocess to get…

```
double x = 3.1415926 * d;
```

COLLEGE OF COMPUTING AND INFORMATICS

Steps in Compiling (Review)



```
#define N 3
a=c+b*N;
```

```
a=c+b*3;
```

```
a  =  c + b * 3  ;
```

Uses of Preprocessing

1. (header) **file inclusion**
   (e.g., `#include <stdio.h>`)

2. **macro substitution** for common (short) fragments of code
   (e.g., `#define PI 3.1415926`)

3. **conditional compilation**
   (e.g., `#ifdef DEBUG` … `#endif`)

COLLEGE OF COMPUTING AND INFORMATICS

Preprocessor Commands

- Any line starting with the **#** character
- A preprocessing command is terminated by the end of the line, **unless** continued with a \
- Ex.:

```
#define PISHORT 3.1416

#define PILONG \
       3.14159265358979323846264
```

COLLEGE OF COMPUTING AND INFORMATICS

#define

#define identifier token-sequence

- Preprocessor: anywhere it finds **identifier** in the program, it replaces it with **token-sequence**
- One use: giving names to “magic” numbers.
  For example:

```
#define E 2.718282
#define BIGRAISE 50000
#define FALSE 0
#define TRUE 1
#define ERROR -1
#define EQ ==
#define TABSIZE 100
```

#define (cont'd)

```
if (really_good_year EQ TRUE)
    salary += BIGRAISE;
```

preprocess to get…

```
if (really_good_year == 1)
    salary += 50000;
```

- This is **not** the same as declaring a variable; no storage is allocated

#define (cont'd)

```
#define E 2.718282
#define BIGRAISE 50000
#define FALSE 0
#define TRUE 1
#define ERROR -1
#define EQ ==
#define TABSIZE 100
```

```
int table[TABSIZE];
…
for (i = 0; i < TABSIZE; i++)
    if (table[i] EQ 15)
    …
```

translated by the preprocessor (before compiling) into…

```
int table[100];
…
for (i = 0; i < 100; i++)
    if (table[i] == 15)
    …
```

COLLEGE OF COMPUTING AND INFORMATICS

More About #define

- #defines can also contain #define'd values

```
#define PI 3.1415926
#define TWOPI 2*PI
```

By convention, #define identifiers are written in **ALL CAPS**

Do not terminate #define by `;` or it becomes part of **token_sequence**!

☠ *common source of bugs* ☠
Terminating macro definition with `;`

```
#define PI 3.1416 ;
…
area = PI * r * r;
```

$\rightarrow$

```
area = 3.1416 ; * r * r;
```

COLLEGE OF COMPUTING AND INFORMATICS

`#include`

Inserts into the source code the **contents of another file**

– often called a ***header*** file (filetype: `.h`)

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

standard library header file

user defined header file

Where does **gcc** look for these files?

– installation dependent (but often `/usr/include`)

– same directory as source code file

– other locations controlled by `gcc -I` **option**

COLLEGE OF COMPUTING AND INFORMATICS

`#include` (cont'd)

- Frequently part of header files:
  - constant definitions
  - function prototype declarations (for libraries)
  - **`extern`** declarations
- When the header file changes, all source files that **`#include`** it have to be **recompiled**
  - i.e., there is a **dependency** of this source code on the contents of the header file

Some Useful (Standard) Header Files

- `stdio.h`
- `stddef.h`
- `math.h`
- `string.h`
- `float.h` and `limits.h`
- Take a look in `/usr/include` on your system

Conditional Compilation

- To control what source code gets compiled
- Common uses
  - to resolve, at compile time, **platform** (machine- or OS-) **dependencies**
  - to compile (or not) **debugging code**
- Requires the following preprocessor directives
  - **`#if / #ifdef / #ifndef`**
  - **`#elif / #else`**
  - **`#endif`**

Conditional Compilation: Example

*

```
#if defined(LINUX)
  #define HDR "linux.h"
#elif defined(WIN32)
  #define HDR "windows.h"
#else
  #define HDR "default.h"
#endif

#include HDR
```

And when compiling this program, can define what **SYSTEM** is by using the **-D** option to **gcc**

```
gcc -DWIN32 myprog.c …
```

```
#include "windows.h"
```

(see **system.c** in *Code Samples and Demonstrations* in *Canvas*)

COLLEGE OF COMPUTING AND INFORMATICS

References

- S. J. Matthews, T. Newhall and K. C. Webb, *Dive into Systems*, Version 1.2. Free online textbook, available at: https://diveintosystems.org/book/
- K. N. King, *C Programming: A Modern Approach*, 2nd Edition. W. W. Norton & Company. 2008.
- D.S. Malik, *C++ Programming: From Problem Analysis to Program Design*, Seventh Edition. Cengage Learning. 2014.

COLLEGE OF COMPUTING AND INFORMATICS