Computer Programming II

Ming-Feng Tsai (Victor Tsai)

Dept. of Computer Science
National Chengchi University

C Preprocessors

Conditional Compilation

Conditional Compilation

- Through the use of conditional compilation, the preprocessor allows programmers great flexibility in changing the code generated

Conditional Compilation

- Through the use of conditional compilation, the preprocessor allows programmers great flexibility in changing the code generated
- **`#ifdef / #endif`**

Conditional Compilation

- Through the use of conditional compilation, the preprocessor allows programmers great flexibility in changing the code generated
- **`#ifdef / #endif`**
  - put debugging code in the program while still working on it, and then remove it in the final version

Conditional Compilation

Conditional Compilation

- Example of of **`#ifdef` / `#endif`**

Conditional Compilation

- Example of of **`#ifdef` / `#endif`**

```
#ifdef DEBUG
  printf("In compute_hash, value %d hash %d\n",
  value, hash);
#endif  /* DEBUG */
```

Conditional Compilation

- Example of of **`#ifdef` / `#endif`**

```
#ifdef DEBUG
  printf("In compute_hash, value %d hash %d\n",
  value, hash);
#endif  /* DEBUG */
```

- If the beginning of the program contains:

```
#define DEBUG /* Turn debugging on */
```

the printf will be included. If the program contains:

```
#undef DEBUG /* Turn debugging off */
```

the printf will be omitted

Conditional Compilation

Conditional Compilation

- **`#ifndef`**
  - cause the code to be compiled if the symbol is **not** defined

Conditional Compilation

- **`#ifndef`**
  - cause the code to be compiled if the symbol is **not** defined

```
#ifndef DEBUG
    printf("Production code, no debugging enabled\n");
#endif /* DEBUG */
```

Conditional Compilation

- **`#ifndef`**
  - cause the code to be compiled if the symbol is **not** defined

```
#ifndef DEBUG
    printf("Production code, no debugging enabled\n");
#endif /* DEBUG */
```

- **`#else`**

Conditional Compilation

- **`#ifndef`**
  - cause the code to be compiled if the symbol is **not** defined

```
#ifndef DEBUG
    printf("Production code, no debugging enabled\n");
#endif /* DEBUG */
```

- **`#else`**

```
#ifdef DEBUG
    printf("Test version. Debugging is on\n");
#else DEBUG
    printf("Production version\n");
#endif /* DEBUG */
```

Conditional Compilation

Conditional Compilation

- Remove a section of code temporarily

Conditional Compilation

- Remove a section of code temporarily
  - common method: use comment

Conditional Compilation

- Remove a section of code temporarily
  - common method: use comment
  - better method is to use the **`#ifdef`** directive

Conditional Compilation

- Remove a section of code temporarily
  - common method: use comment
  - better method is to use the **`#ifdef`** directive

```
#ifdef UNDEF
    section_report();
    /* Handle the end of section stuff */
    dump_table();
#endif /* UNDEF */
```

Conditional Compilation

- Remove a section of code temporarily
  - common method: use comment
  - better method is to use the **`#ifdef`** directive

```
#ifdef UNDEF
    section_report();
    /* Handle the end of section stuff */
    dump_table();
#endif /* UNDEF */
```

Conditional Compilation

- Remove a section of code temporarily
  - common method: use comment
  - better method is to use the **`#ifdef`** directive

```
#ifdef UNDEF
    section_report();
    /* Handle the end of section stuff */
    dump_table();
#endif /* UNDEF */
```

should define the symbol UNDEF to compile this section of codes

Conditional Compilation

Conditional Compilation

- Define a symbol

Conditional Compilation

- Define a symbol
  - write in code

Conditional Compilation

- Define a symbol
  - write in code
  - use compiler switch **`-Dsymbol`**

```
gcc -DUNDEF -g -o prog prog.c
```

Conditional Compilation

- Define a symbol
  - write in code
  - use compiler switch **`-Dsymbol`**

```
gcc -DUNDEF -g -o prog prog.c
```

  - another form of the switch **`-Dsymbol=value`**

```
gcc -DMAX=10 -o prog prog.c
```

Conditional Compilation

Conditional Compilation

- So, you can use `-DDEBUG` to turn on all the codes in between **`#ifdef DEBUG`** and **`#endif`** even though there is no `#define DEBUG` in the program

Conditional Compilation

- So, you can use `-DDEBUG` to turn on all the codes in between **`#ifdef DEBUG`** and **`#endif`** even though there is no `#define DEBUG` in the program
- Notice that the programmer can override the command-line options with directives in the program. For example

```
#undef DEBUG
```

Conditional Compilation

- Example: debug.c

```
3  #ifdef USA
4  #define currency_rate 46
5  #endif
6
7  int main()
8  {
9      int rs;
10     rs = 10 * currency_rate;
11     printf ("%d\n", rs);
12
13     #ifdef DEBUG
14         printf("Debug message!!\n");
15     #endif
16
17     return (0);
18 }
```

Conditional Compilation

- Example: debug.c

```
3  #ifdef USA
4  #define currency_rate 46
5  #endif
6
7  int main()
8  {
9      int rs;
10     rs = 10 * currency_rate;
11     printf ("%d\n", rs);
12
13     #ifdef DEBUG
14         printf("Debug message!!\n");
15     #endif
16
17     return (0);
18 }
```

Conditional Compilation

- Example: debug.c

```
3  #ifdef USA
4  #define currency_rate 46
5  #endif
6
7  int main()
8  {
9      int rs;
10     rs = 10 * currency_rate;
11     printf ("%d\n", rs);
12
13     #ifdef DEBUG
14         printf("Debug message!!\n");
15     #endif
16
17     return (0);
18 }
```

Conditional Compilation

- Example: debug.c

```
3  #ifdef USA
4  #define currency_rate 46
5  #endif
6
7  int main()
8  {
9      int rs;
10     rs = 10 * currency_rate;
11     printf ("%d\n", rs);
12
13     #ifdef DEBUG
14         printf("Debug message!!\n");
15     #endif
16
17     return (0);
18 }
```

```
gcc -DUSA debug.c
```

Conditional Compilation

- Example: debug.c

```
3  #ifdef USA
4  #define currency_rate 46
5  #endif
6
7  int main()
8  {
9      int rs;
10     rs = 10 * currency_rate;
11     printf ("%d\n", rs);
12
13     #ifdef DEBUG
14         printf("Debug message!!\n");
15     #endif
16
17     return (0);
18 }
```

```
gcc -DUSA debug.c
```

```
460
```

Conditional Compilation

- Example: debug.c

```
3  #ifdef USA
4  #define currency_rate 46
5  #endif
6
7  int main()
8  {
9      int rs;
10     rs = 10 * currency_rate;
11     printf ("%d\n", rs);
12
13     #ifdef DEBUG
14         printf("Debug message!!\n");
15     #endif
16
17     return (0);
18 }
```

```
gcc -DUSA debug.c
```

```
460
```

```
gcc -DUSA -DDEBUG debug.c
```

Conditional Compilation

- Example: debug.c

```
3  #ifdef USA
4  #define currency_rate 46
5  #endif
6
7  int main()
8  {
9      int rs;
10     rs = 10 * currency_rate;
11     printf ("%d\n", rs);
12
13     #ifdef DEBUG
14         printf("Debug message!!\n");
15     #endif
16
17     return (0);
18 }
```

```
gcc -DUSA debug.c
```

```
460
```

```
gcc -DUSA -DDEBUG debug.c
```

```
460
Debug message!!
```