

## C Pre-processor

Alon Efrat  
Computer Science Department  
University of Arizona

## #include

- #include <stdio.h> start looking for this file using the path
- #include "filename" start looking in current dir
- #include <filename>
  
- The #include ... line is replaced by the contents of the file included.
- A preferred way to tie the declarations together for a large program.

2

## #define

- #define name
- #define name text
- #define PI 3.1415926 ; #define e 2.17 **ERROR**
- #define TRUE 0
- #define AND &&
- #define copyright All right reserved. No part \\ of this publication may be reproduce, \\ stored in a retrieval system
- #define record struct
  
- Macro substitution: **literally** replace occurrences of name in the program with text
- name has the same form as variables

3

## #define - cont.

- Usage:
  - Constants
  - Repetitive small block of code
  - Debug code
  
- #undef name
  - It is not an error if the name is not defined originally

4

## #define (cont.)

- #define name(parameters) text
- Examples:
  - #define MAX(a, b) (a>b?a: b)  
MAX( i++,j++ ) ==> i++ > j++ ? i++ : j++
  - #define square(x) x\*x **Wrong**  
square(x+2) ==> x+2 \* x+2
  - #define square(x) (x)\*(x) **ok**

5

## #define (cont.)

- #define name(parameters) text
- #parameter in the text will put the parameter in double quote
- #define dprint(exp) printf(#exp " = %g\n ", exp )
  - dprint( x/y ) ==> printf( "x/y" " =%g\n", x/y )
  - dprint( 2/3 ) **output:** 2/3 = 0.666

6

## Another example

```
#define PRINT_SIZE(type) \
printf("%d\t:" #type "\n", \
sizeof(type))

int main(void)
{
    PRINT_SIZE(char);
    PRINT_SIZE(unsigned char);
    PRINT_SIZE(int);
    PRINT_SIZE(short int);
    PRINT_SIZE(long int);
    PRINT_SIZE(long long int);
    PRINT_SIZE(long);
    PRINT_SIZE(float);
    PRINT_SIZE(double);
    PRINT_SIZE(long double);
    return 0;
}
```

7

## Conditional Code

```
#ifdef name
<code>
#endif
```

```
#ifdef name
<code>
#else
<code>
#endif
```

```
#ifndef name
<code>
#endif
```

```
#ifndef name
<code>
#else
<code>
#endif
```

- Making portable code
- Prevent multi-inclusion of header files
- Debugging

8

## Conditional Code (cont.)

```
#if constant_expression_1
<code 1>
#elif constant_expression_2
<code 2>
#else
<code 3>
#endif
```

9

## Predefined Names

- `__LINE__`: current source line number
- `__FILE__`: filename of the file currently being compiled
- Used for debugging

10

## Passing macros in compilation

- `gcc file8.c -Dkuku=17 -o file8`

```
main(){
    printf("%d%", kuku);
}
```

11

## Acknowledgement

- John H. Hartman, *Classnotes for Csc352-Spring03*, CS Dept., University of Arizona, 2003
- Brian W. Kernighan, Dennis M. Ritchie, *The C Programming Language (2<sup>nd</sup> Ed.)*, Prentice Hall, 1988

12