If I have seen further, it is by standing on the shoulders of giants.

--- *Isaac Newton*

C 程 序 设 计

C Programming

16

C预处理器和库

理论课程

厦門大學

XIAMEN UNIVERSITY

信息学院 黄 煒

（国家示范性软件学院） 博士，副教授

School of Informatics Dr. Wei Huang

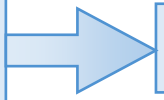
知识框架

- 预处理
  - #include
  - #define
  - 其它指令：#if #ifdef #ifndef #else #elif #endif
- 库函数

处理程序的第一步

- 先把源代码中出现的字符映射到源字符集
  - 多字节字符、三元字符扩展等
- 查找反斜杠紧跟换行符的示例，并合并

```
printf("That's wond\
erful!\n");
```



```
printf("That's wonderful!\n");
```

- 编译器将文本划分成预处理的语言符号序列和空白字符序列及注释序列
  - 编译器用一个空白字符代替每一个注释
  - 用单个空格替代每一个空白字符（除换行符）

预处理

- 预编译器寻找可能存在的预处理指令
  - 预处理指令在一行的开始处以“#”标识
  - ANSI C允许#前后有空格或制表符，但旧版C不允许

```
# define MAX 32767
```

  - 预编译自#开始，至换行符结束
    - 处理换行符在预处理之前，因而斜杠紧跟换行符、注释均不受影响

```c
/* preproc.c -- simple preprocessor examples */
#include <stdio.h>
#define TWO 2        /* you can use comments if you like   */
#define OW "Consistency is the last refuge of the unimagina\
tive. - Oscar Wilde" /* a backslash continues a definition */
/* to the next line                          */
#define FOUR  TWO*TWO
#define PX printf("X is %d.\n", x)
#define FMT  "X is %d.\n"
int main(void)
{
    int x = TWO;
    PX;
    x = FOUR;
    printf(FMT, x);
    printf("%s\n", OW);
    printf("TWO: OW\n");
    return 0;
}
```

```
X is 2.
X is 4.
Consistency is the last refuge of the
unimaginative. - Oscar Wilde
TWO: OW
```

明显常量 #define

- 每个#define行（逻辑行）由三个部分组成

```
#define PX printf("X is %d.\n", x)
```

预编译指令 宏 主体

  - 预编译指令：#define
  - 宏（macro）：所选择的缩略语
    - 类对象宏（object-like macro）：用宏代表值
    - 只能是字母、数字或者下划线，且不以数字开头
  - 主体（body）：替换列表（replacement list）
- 预处理器发现宏示例后总会用主体替代宏
  - 该过程称为宏展开（macro expansion）

明显常量 #define

- 宏定义可以包含其他宏（不包括自己）
  - 有些编译器不支持该功能
- 字符串中的宏不被替换

语言符号

- 系统将宏的主体当做语言符号（token）类型字符串
- C处理器中的语言符号是宏定义主体中单独的词（word），用空白将词分开
- 在C预编译阶段，“2 * 2”是三个符号，而“2*2”是一个符号
- 但在编译阶段，这些都是三个符号

重定义常量

- 以下认为是相同的预定义

```
#define X 2 * 2
#define X 2    *    2
```

- 以下认为发生了重定义

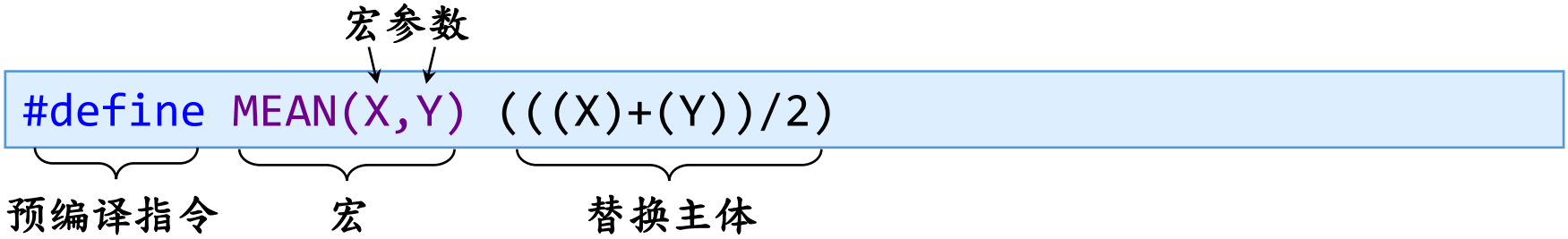
```
#define X 2 * 2
#define X 2*2
```

```
#define X 2 * 2
#define X 2 * 3
```

```
a.c:5:0: warning: "X" redefined [enabled by default]
 #define X 2*2
 ^
a.c:4:0: note: this is the location of the previous definition
 #define X 2 *    2
```

#define中使用参数

- 类函数宏（function-like macro）



- 宏展开是预编译范畴，不是编译器范畴
  - 不同宏定义对宏调用245/SQUARE(X+2)的展开

预定义宏	宏展开	值（X为5时）
#define SQUARE(X) X*X	245/X+2*X+2	245/5+2*5+2=61
#define SQUARE(X) (X)*(X)	245/(X+2)*(X+2)	245/(5+2)*(5+2)=245
#define SQUARE(X) ((X)*(X))	245/((X+2)*(X+2))	245/((5+2)*(5+2))=5

  - 应对宏参数和替换列表分别加括号避免优先级错误

```c
/* mac_arg.c -- macros with arguments */
#include <stdio.h>
#define SQUARE(X) X*X
#define PR(X)   printf("The result is %d.\n", X)

int main(void)
{
    int x = 5;
    int z;

    printf("x = %d\n", x);
    z = SQUARE(x);
    printf("Evaluating SQUARE(x): ");
    PR(z);
    z = SQUARE(2);
    printf("Evaluating SQUARE(2): ");
    PR(z);
    printf("Evaluating SQUARE(x+2): ");
```

```
    PR(SQUARE(x+2));
    printf("Evaluating 100/SQUARE(2): ");
    PR(100/SQUARE(2));
    printf("x is %d.\n", x);
    printf("Evaluating SQUARE(++x): ");
    PR(SQUARE(++x));
    printf("After incrementing, x is %x.\n", x);

    return 0;
}
```

```
x = 5
Evaluating SQUARE(x): The result is 25.
Evaluating SQUARE(2): The result is 4.
Evaluating SQUARE(x+2): The result is 17.
Evaluating 100/SQUARE(2): The result is 100.
x is 5.
Evaluating SQUARE(++x): The result is 49.
After incrementing, x is 7.
```

#define中使用参数

- 使用宏参数创建字符串：#运算符

```
#define PSQR(x) printf("The square of " #x " is %d.\n",((x)*(x)))
```

  - 例如：用PSQR(2 + 4)调用宏时，#x将被"2 + 4"替代

- 预处理器粘合剂

```
#define PRINT_XN(n) printf("x" #n " = %d\n", x ## n);
```

  - 将两个语言符号组成一个语言符号
    - 如果该语言符号可以继续宏展开，将继续宏展开

```c
/* subst.c -- substitute in string */
#include <stdio.h>
#define PSQR(x) printf("The square of " #x " is
%d.\n",((x)*(x)))

int main(void)
{
    int y = 5;

    PSQR(y);
    PSQR(2 + 4);

    return 0;
}
```

```
The square of y is 25.
The square of 2 + 4 is 36.
```

```c
// glue.c -- use the ## operator
#include <stdio.h>
#define XNAME(n) x ## n
#define PRINT_XN(n) printf("x" #n " = %d\n", x ## n);

int main(void)
{
    int XNAME(1) = 14;  // becomes int x1 = 14;
    int XNAME(2) = 20;  // becomes int x2 = 20;
    int x3 = 30;
    PRINT_XN(1);        // becomes printf("x1 = %d\n", x1);
    PRINT_XN(2);        // becomes printf("x2 = %d\n", x2);
    PRINT_XN(3);        // becomes printf("x3 = %d\n", x3);
    return 0;
}
```

```
x1 = 14
x2 = 20
x3 = 30
```

#define中使用参数

- 可变宏：...和__VA_ARGS__

```
#define PR(X, ...) printf("Message " #X ": " __VA_ARGS__)
```

- 这是应对有些函数接受可变数量的参数
- 同样地，可变宏应是最后一个参数

```c
// variadic.c -- variadic macros
#include <stdio.h>
#include <math.h>
#define PR(X, ...) printf("Message " #X ": " __VA_ARGS__)

int main(void)
{
    double x = 48;
    double y;

    y = sqrt(x);
    PR(1, "x = %g\n", x);
    PR(2, "x = %.2f, y = %.4f\n", x, y);

    return 0;
}
```

```
Message 1: x = 48
Message 2: x = 48.00, y = 6.9282
```

宏还是函数？

- 宏：稍有不慎会产生奇怪现象
  - 考虑兼容其他编译器，宏应尽量写在一行
- 宏与函数实际上是时间与空间的权衡
  - 宏产生内联代码（C99提供了第三种方法：内联函数）
  - 函数只有一个副本，但需要控制权转换
- 注意
  - 宏名不含空格，括号括住参数与整体定义，大写字母
  - 先确定使用宏会不会更省时？考虑只调用一次的宏

文件包含#include

- #include将其指定的文件名包含到文件中
- #include的两种形式
  - #include <stdio.h>　　（搜索系统目录）
  - #include "my.h"　　（搜索当前目录）
  - #include "/usr/my.h"　　（搜索/usr/目录）
- 包含头文件并不会显著增加程序的大小
  - 最终无关的部分不会进去可执行程序

文件包含#include

- 头文件放什么
  - 明显常量
  - 宏函数
  - 函数声明
  - 结构模板定义
  - 类型定义
  - 可以使用头文件声明多个文件共享的外部变量

```
// names_st.h -- names_st structure header file
// constants
#include <string.h>
#define SLEN 32

// structure declarations
struct names_st
{
    char first[SLEN];
    char last[SLEN];
};

// typedefs
typedef struct names_st names;

// function prototypes
void get_names(names *);
void show_names(const names *);
char * s_gets(char * st, int n);
```

```c
// names_st.c -- define names_st functions
#include <stdio.h>
#include "names_st.h"     // include the header file

// function definitions
void get_names(names * pn)
{
    printf("Please enter your first name: ");
    s_gets(pn->first, SLEN);

    printf("Please enter your last name: ");
    s_gets(pn->last, SLEN);
 }

void show_names(const names * pn)
{
    printf("%s %s", pn->first, pn->last);
}
```

```c
char * s_gets(char * st, int n)
{
    char * ret_val;
    char * find;

    ret_val = fgets(st, n, stdin);
    if (ret_val)
    {
        find = strchr(st, '\n'); // look for newline
        if (find)                // if the address is not NULL,
            *find = '\0';        // place a null character there
        else
            while (getchar() != '\n')
                continue;        // dispose of rest of line
    }
    return ret_val;
}
```

```c
// useheader.c -- use the names_st structure
#include <stdio.h>
#include "names_st.h"
// remember to link with names_st.c

int main(void)
{
    names candidate;

    get_names(&candidate);
    printf("Let's welcome ");
    show_names(&candidate);
    printf(" to this program!\n");
    return 0;
}
```

```
Please enter your first name: Wei↲
Please enter your last name: Huang↲
Let's welcome Wei Huang to this program!
```

其它指令

扩展内容

- 取消定义指令：#undef
  - 注意：一些宏不可以取消定义
- 条件编译：#ifdef … #else … #endif
  - 也可以使用#ifndef
  - 也可以使用#elif（早期不支持）
- 预定义宏（如时间、文件名等）
- 重置行号和文件名：#line

```c
/* ifdef.c -- uses conditional compilation */
#include <stdio.h>
#define JUST_CHECKING
#define LIMIT 4
int main(void)
{
    int i;
    int total = 0;
    for (i = 1; i <= LIMIT; i++)
    {
        total += 2*i*i + 1;
#ifdef JUST_CHECKING
        printf("i=%d, running total = %d\n", i, total);
#endif
    }
    printf("Grand total = %d\n", total);
    return 0;
}
```

```
i=1, running total = 3
i=2, running total = 12
i=3, running total = 31
i=4, running total = 64
Grand total = 64
```

```
// names.h --revised with include protection
#ifndef NAMES_H_
#define NAMES_H_

// constants
#define SLEN 32
// structure declarations
struct names_st
{
    char first[SLEN];
    char last[SLEN];
};
// typedefs
typedef struct names_st names;
// function prototypes
void get_names(names *);
void show_names(const names *);
char * s_gets(char * st, int n);
#endif
```

```c
// doubincl.c -- include header twice
#include <stdio.h>
#include "names.h"
#include "names.h"   // accidental second inclusion

int main()
{
    names winner = {"Less", "Ismoor"};
    printf("The winner is %s %s.\n", winner.first,
           winner.last);
    return 0;
}
```

```
The winner is Less Ismoor.
```

```c
// predef.c -- predefined identifiers
#include <stdio.h>
void why_me();
int main()
{
    printf("The file is %s.\n", __FILE__);
    printf("The date is %s.\n", __DATE__);
    printf("The time is %s.\n", __TIME__);
    printf("The version is %ld.\n", __STDC_VERSION__);
    printf("This is line %d.\n", __LINE__);
    printf("This function is %s\n", __func__);
    why_me();
    return 0;
}
void why_me()
{
    printf("This function is %s\n", __func__);
    printf("This is line %d.\n", __LINE__);
}
```

```
dev@ubuntu:~/Desktop$ gcc predef.c -o predef -std=c11
```

```
The file is predef.c.
The date is Aug 19 2014.
The time is 22:17:00.
The version is 201112.
This is line 11.
This function is main
This function is why_me
This is line 21.
```

其它指令

扩展内容

- 预处理器错误指令：#error
- 编译指示：#pragma
  - 每个编译器都有自己的编译指示集
  - C99提供了_Pragma预处理器运算符

内联函数：inline

扩展内容

- 声明函数为内联将简化调用机制，但也可能不起作用
- 使用内联函数前应先定义（不能只是声明原型）
- 不能取地址
  - 一旦取地址则丧失内联
- 应针对短小的程序
  - 太长的程序调用并不花多少时间，不起作用
- 一般不在头文件定义函数，内联函数除外

C库

扩展内容

- 没有标准→基于UNIX的C为事实标准→ANSI C
- 系统使用不同的方法搜索C库的函数
  - 自动访问：常见的库
  - 文件包含：如：宏
  - 库包含：不常用的函数库
- 参考库描述

数学库

扩展内容

- 头文件：math.h
- UNIX使用-lm标记只是连接搜索数学库

```c
/* rect_pol.c -- converts rectangular coordinates to polar */
#include <stdio.h>
#include <math.h>

#define RAD_TO_DEG (180/(4 * atan(1)))

typedef struct polar_v {
    double magnitude;
    double angle;
} Polar_V;

typedef struct rect_v {
    double x;
    double y;
} Rect_V;

Polar_V rect_to_polar(Rect_V);
```

```c
int main(void)
{
    Rect_V input;
    Polar_V result;
    puts("Enter x and y coordinates; enter q to quit:");
    while (scanf("%lf %lf", &input.x, &input.y) == 2)
    {
        result = rect_to_polar(input);
        printf("magnitude = %0.2f, angle = %0.2f\n",
               result.magnitude, result.angle);
    }
    puts("Bye.");

    return 0;
}
```

```
Polar_V rect_to_polar(Rect_V rv)
{
    Polar_V pv;
    pv.magnitude = sqrt(rv.x * rv.x + rv.y * rv.y);
    if (pv.magnitude == 0)
        pv.angle = 0.0;
    else
        pv.angle = RAD_TO_DEG * atan2(rv.y, rv.x);
    return pv;
}
```

```
Enter x and y coordinates; enter q to quit:
23 43↲
magnitude = 48.76, angle = 61.86
65 9↲
magnitude = 65.62, angle = 7.88
q↲
Bye.
```

通用工具库

扩展内容

- exit()和atexit()函数
- qsort()函数
- 请自习

```c
//  generic.c  -- defining generic macros

#include <stdio.h>
#include <math.h>
#define RAD_TO_DEG (180/(4 * atanl(1)))

// generic square root function
#define SQRT(X) _Generic((X),\
    long double: sqrtl, \
    default: sqrt, \
    float: sqrtf)(X)

// generic sine function, angle in degrees
#define SIN(X) _Generic((X),\
long double: sinl((X)/RAD_TO_DEG),\
default:     sin((X)/RAD_TO_DEG),\
float:       sinf((X)/RAD_TO_DEG)\
)
```

```c
int main(void)
{
    float x = 45.0f;
    double xx = 45.0;
    long double xxx =45.0L;
    long double y = SQRT(x);
    long double yy= SQRT(xx);
    long double yyy = SQRT(xxx);
    printf("%.17Lf\n", y);   // matches float
    printf("%.17Lf\n", yy);  // matches default
    printf("%.17Lf\n", yyy); // matches long double
    int i = 45;
    yy = SQRT(i);            // matches default
    printf("%.17Lf\n", yy);
    yyy= SIN(xxx);           // matches long double
    printf("%.17Lf\n", yyy);
    return 0;
}
```

```c
/* byebye.c -- atexit() example */
#include <stdio.h>
#include <stdlib.h>
void sign_off(void);
void too_bad(void);
int main(void)
{
    int n;
    atexit(sign_off);    /* register the sign_off() function */
    puts("Enter an integer:");
    if (scanf("%d",&n) != 1)
    {
        puts("That's no integer!");
        atexit(too_bad); /* register the too_bad()  function */
        exit(EXIT_FAILURE);
    }
    printf("%d is %s.\n", n,  (n % 2 == 0)? "even" : "odd");
    return 0;
}
```

```c
void sign_off(void)
{
    puts("Thus terminates another magnificent program from");
    puts("SeeSaw Software!");
}

void too_bad(void)
{
    puts("SeeSaw Software extends its heartfelt condolences");
    puts("to you upon the failure of your program.");
}
```

```
Enter an integer:
56↲
56 is even.
Thus terminates another magnificent program from
SeeSaw Software!
```

```c
/* qsorter.c -- using qsort to sort groups of numbers */
#include <stdio.h>
#include <stdlib.h>
#define NUM 40
void fillarray(double ar[], int n);
void showarray(const double ar[], int n);
int mycomp(const void * p1, const void * p2);

int main(void)
{
    double vals[NUM];
    fillarray(vals, NUM);
    puts("Random list:");
    showarray(vals, NUM);
    qsort(vals, NUM, sizeof(double), mycomp);
    puts("\nSorted list:");
    showarray(vals, NUM);
    return 0;
}
```

```
void fillarray(double ar[], int n)
{
    int index;
    for( index = 0; index < n; index++)
        ar[index] = (double)rand()/((double) rand() + 0.1);
}

void showarray(const double ar[], int n)
{
    int index;
    for( index = 0; index < n; index++)
    {
        printf("%9.4f ", ar[index]);
        if (index % 6 == 5)
            putchar('\n');
    }
    if (index % 6 != 0)
        putchar('\n');
}
```

```c
/* sort by increasing value */
int mycomp(const void * p1, const void * p2)
{
    /* need to use pointers to double to access values   */
    const double * a1 = (const double *) p1;
    const double * a2 = (const double *) p2;

    if (*a1 < *a2)
        return -1;
    else if (*a1==*a2)
        return 0;
    else
        return 1;
}
```

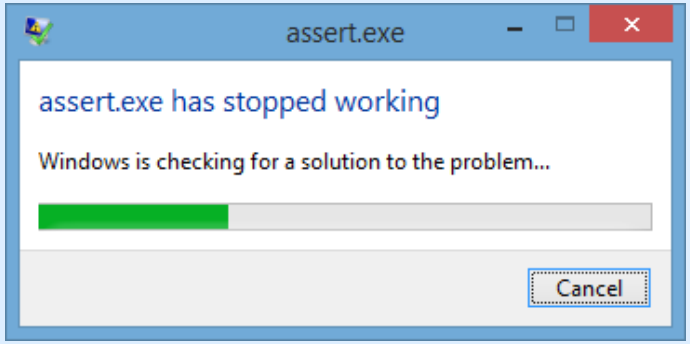
```
Random list:
   0.0022    0.2390    1.2191    0.3910    1.1021    0.2027
   1.3835   20.2830    0.2508    0.8880    2.2179   25.4866
   0.0236    0.9308    0.9911    0.2507    1.2802    0.0939
   0.9760    1.7217    1.2054    1.0326    3.7892    1.9635
   4.1137    0.9241    0.9971    1.5582    0.8955   35.3798
   4.0579   12.0460    0.0096    1.0109    0.8506    1.1529
   2.3614    1.5876    0.4825    6.8749

Sorted list:
   0.0022    0.0096    0.0236    0.0939    0.2027    0.2390
   0.2507    0.2508    0.3910    0.4825    0.8506    0.8880
   0.8955    0.9241    0.9308    0.9760    0.9911    0.9971
   1.0109    1.0326    1.1021    1.1529    1.2054    1.2191
   1.2802    1.3835    1.5582    1.5876    1.7217    1.9635
   2.2179    2.3614    3.7892    4.0579    4.1137    6.8749
  12.0460   20.2830   25.4866   35.3798
```

诊断库

扩展内容

- 头文件：assert.h
- 请自习

```c
/* assert.c -- use assert() */
#include <stdio.h>
#include <math.h>
#include <assert.h>
int main()
{
    double x, y, z;
    puts("Enter a pair of numbers (0 0 to quit): ");
    while (scanf("%lf%lf", &x, &y) == 2
           && (x != 0 || y != 0))
    {
        z = x * x - y * y;  /* should be + */
        assert(z >= 0);
        printf("answer is %f\n", sqrt(z));
        puts("Next pair of numbers: ");
    }
    puts("Done");
    return 0;
}
```




```
Enter a pair of numbers (0 0 to quit):
32↲
39↲
Assertion failed: z >= 0, file assert.c, line 14
```

```
Enter a pair of numbers (0 0 to quit):
6 3↲
answer is 5.196152
Next pair of numbers:
0 0↲
Done
```

```c
//  statasrt.c
#include <stdio.h>
#include <limits.h>
_Static_assert(CHAR_BIT == 16, "16-bit char falsely assumed");
int main(void)
{
    puts("char is 16 bits.");
    return 0;
}
```

```
gcc statasrt.c↵
statasrt.c:4:1: error: static assertion failed: "16-bit
char falsely assumed"
 _Static_assert(CHAR_BIT == 16, "16-bit char falsely
assumed");
```

字符串库的memcpy()和memmove()

扩展内容

- 请自习

```c
// mems.c -- using memcpy() and memmove()
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define SIZE 10
void show_array(const int ar[], int n);
// remove following if C11 _Static_assert not supported
_Static_assert(sizeof(double) == 2 * sizeof(int), "double not
twice int size");

int main()
{
    int values[SIZE] = {1,2,3,4,5,6,7,8,9,10};
    int target[SIZE];
    double curious[SIZE / 2] = {2.0, 2.0e5, 2.0e10, 2.0e20,
5.0e30};

    puts("memcpy() used:");
```

```
    puts("values (original data): ");
    show_array(values, SIZE);
    memcpy(target, values, SIZE * sizeof(int));
    puts("target (copy of values):");
    show_array(target, SIZE);
    puts("\nUsing memmove() with overlapping ranges:");
    memmove(values + 2, values, 5 * sizeof(int));
    puts("values -- elements 0-5 copied to 2-7:");
    show_array(values, SIZE);

    puts("\nUsing memcpy() to copy double to int:");
    memcpy(target, curious, (SIZE / 2) * sizeof(double));
    puts("target -- 5 doubles into 10 int positions:");
    show_array(target, SIZE/2);
    show_array(target + 5, SIZE/2);

    return 0;
}
```

```c
void show_array(const int ar[], int n)
{
    int i;

    for (i = 0; i < n; i++)
        printf("%d ", ar[i]);
    putchar('\n');
}
```

```
dev@ubuntu:~/Desktop$ gcc mems.c↵
dev@ubuntu:~/Desktop$ ./a.out↵
memcpy() used:
values (original data):
1 2 3 4 5 6 7 8 9 10
target (copy of values):
1 2 3 4 5 6 7 8 9 10

Using memmove() with overlapping ranges:
values -- elements 0-5 copied to 2-7:
1 2 1 2 3 4 5 8 9 10

Using memcpy() to copy double to int:
target -- 5 doubles into 10 int positions:
0 1073741824 0 1091070464 536870912
1108516959 2025163840 1143320349 -2012696540 1179618799
```

可变参数stdarg.h

扩展内容

- 请自习

```c
//varargs.c -- use variable number of arguments
#include <stdio.h>
#include <stdarg.h>
double sum(int, ...);

int main(void)
{
    double s,t;

    s = sum(3, 1.1, 2.5, 13.3);
    t = sum(6, 1.1, 2.1, 13.1, 4.1, 5.1, 6.1);
    printf("return value for "
           "sum(3, 1.1, 2.5, 13.3):                  %g\n", s);
    printf("return value for "
           "sum(6, 1.1, 2.1, 13.1, 4.1, 5.1, 6.1): %g\n", t);

    return 0;
}
```

```
double sum(int lim,...)
{
    va_list ap;             // declare object to hold arguments
    double tot = 0;
    int i;

    va_start(ap, lim);     // initialize ap to argument list
    for (i = 0; i < lim; i++)
        tot += va_arg(ap, double); // access each item in
argument list
    va_end(ap);                    // clean up

    return tot;
}
```

```
return value for sum(3, 1.1, 2.5, 13.3):                    16.9
return value for sum(6, 1.1, 2.1, 13.1, 4.1, 5.1, 6.1): 31.6
```

C 程 序 设 计

C Programming

16

谢谢观看

理论课程

厦門大學

XIAMEN UNIVERSITY

信息学院 黄 炜

（国家示范性软件学院） 博士，副教授

School of Informatics Dr. Wei Huang