



西北工业大学  
NORTHWESTERN POLYTECHNICAL UNIVERSITY

---

# C程序设计 Programming in C



**1011014**

---

主讲：姜学锋，计算机学院

## 调用函数 - 原型与声明

- ◆ 3、常用库函数
- ◆ 4、调用第三方函数

### 4.3.3 标准库函数

---

- ▶ C语言标准提供了一个遵循标准的编译器必须提供的库函数列表，它们是标准所规定的辅助和实用函数，提供基本的或有用的功能，例如数学、输入输出、字符串和时间日期标准库等，而每个标准库中又包括几十到上百个的具体函数。分类如下：

### 4.3.3 标准库函数

表4-1 标准库函数索引

标准库名称	头文件名	标准库名称	头文件名
断言验证	<assert.h>	复数算术运算	<complex.h>
字符类型	<ctype.h>	出错码	<errno.h>
浮点环境	<fenv.h>	浮点常量	<float.h>
整型格式转换	<inttypes.h>	替代记号	<iso646.h>
整型大小	<limits.h>	本地化	<locale.h>
数学	<math.h>	非局部跳转	<setjmp.h>
信号量处理	<signal.h>	可变参数	<stdarg.h>
布尔类型	<stdbool.h>	标准定义	<stddef.h>
整型类型	<stdint.h>	标准输入输出	<stdio.h>
实用函数	<stdlib.h>	字符串	<string.h>
通用类型数学宏	<tgmath.h>	时间日期	<time.h>

### 4.3.3 标准库函数

续表4-1 标准库函数索引

标准库名称	头文件名	标准库名称	头文件名
扩展多字节和宽字符	<wchar.h>	宽字符分类和映射	<wctype.h>

### 4.3.3 标准库函数

---

- ▶ 通常，编译器支持绝大多数的标准库，但也有一些未曾实现。考虑到通用性，本书仅列出常用的函数，如果在编程时需要更多的库和函数，请查阅详细的标准库手册。
- ▶ 在调用标准库函数时，需要在源文件中包含相应的头文件，形式如下：

```
#include <头文件名>
```

### 4.3.3 标准库函数

---

- ▶ 1. 数学库
- ▶ 大部分常用的数学函数都定义在数学库中，其头文件为 `math.h`。

### 4.3.3 标准库函数

---

#### ► (1) acos函数

函数原型: `double acos(double x);`

函数说明: 返回以弧度表示的反余弦值。x要求在 $[-1, +1]$ 区间, 返回值在 $[0, \pi]$ 区间。

应用举例: `y=acos(0.32696); //y=1.237711`



### 4.3.3 标准库函数

---

#### ► (2) asin函数

函数原型: `double asin(double x);`

函数说明: 返回以弧度表示的反正弦值。x要求在 $[-1, +1]$ 区间, 返回值在 $[-\pi/2, \pi/2]$ 区间。

应用举例: `y=asin(0.32696); // y=0.333085`

### 4.3.3 标准库函数

---

#### ▶ (3) atan函数

函数原型: `double atan(double x);`

函数说明: 返回以弧度表示的反正切值。返回值在 $[-\pi/2, \pi/2]$ 区间。

应用举例: `y=atan(-862.42); //y=-1.569637`

### 4.3.3 标准库函数

---

#### ▶ (4) cos函数

函数原型: `double cos(double x);`

函数说明: 返回x的余弦值。x要求为弧度单位。

应用举例: `y=cos(3.1415926535/2); //y=0.0`

### 4.3.3 标准库函数

---

#### ► (5) sin函数

函数原型: `double sin(double x);`

函数说明: 返回x的正弦值。x要求为弧度单位。

应用举例: `y=sin(3.1415926535/2); //y=1.0`

### 4.3.3 标准库函数

---

#### ▶ (6) tan函数

函数原型: `double tan(double x);`

函数说明: 返回x的正切值。x要求为弧度单位。

应用举例: `y=tan(3.1415926535/4); //y=1.0`

### 4.3.3 标准库函数

---

#### ▶ (7) cosh函数

函数原型: `double cosh(double x);`

函数说明: 返回x的双曲余弦值。

应用举例: `y=cosh(3.1415926535/2); //y=2.509178`

### 4.3.3 标准库函数

---

#### ▶ (8) sinh函数

函数原型: `double sinh(double x);`

函数说明: 返回x的双曲正弦值。

应用举例: `y=sinh(3.1415926535/2); //y=2.301299`

### 4.3.3 标准库函数

---

#### ► (9) tanh函数

函数原型: `double tanh(double x);`

函数说明: 返回x的双曲正切值。

应用举例: `y=tanh(1.0); //y=0.761594`



### 4.3.3 标准库函数

---

#### ▶ (10) exp函数

函数原型: `double exp(double x);`

函数说明: 返回e的x次方  $e^x$ 。

应用举例: `y=exp(1.0); //y=2.718282`

### 4.3.3 标准库函数

---

#### ▶ (11) log函数

函数原型: `double log(double x);`

函数说明: 返回x的自然对数。x要求大于0。

应用举例: `y=log(10.0); //y=2.302585`

### 4.3.3 标准库函数

---

#### ▶ (12) log10函数

函数原型: `double log10(double x);`  
函数说明: 返回x以10为底的对数。x要求大于0。  
应用举例: `y=log10(100.0); //y=2.0`

### 4.3.3 标准库函数

---

#### ▶ (13) fabs函数

函数原型: `double fabs(double x);`

函数说明: 返回x的绝对值。

应用举例: `y=fabs(-4.0); //y=4.0`

### 4.3.3 标准库函数

---

#### ▶ (14) pow函数

函数原型: `double pow(double x, double y);`

函数说明: 返回x的y次方  $x^y$ 。若x为负则y必须是整数, 若x为0则y必须大于0。

应用举例: `y=pow(4.0, 4.0); // y=256.0`

### 4.3.3 标准库函数

---

#### ▶ (15) sqrt函数

函数原型: `double sqrt(double x);`

函数说明: 返回x的平方根  $\sqrt{x}$ 。x要求大于等于0。

应用举例: `y=sqrt(9.0); //y=3.0`

### 4.3.3 标准库函数

---



#### 【例4.4】

---

输出 $[0,90)$ 区间的正弦表，每隔 $0.1^\circ$  输出一个正弦值。

### 4.3.3 标准库函数

#### 例4.4

```
1 #include <stdio.h>
2 #include <math.h> //使用数学库
3 int main()
4 {
5     double d;
6     int i,j;
7     for(i=0;i<90;i++) {
8         printf("%2d ",i);
9         for(j=0;j<10;j++) {
10             d=(i+j/10.0)*3.1415926535/180; //角度转换为弧度
11             printf("%.4lf ",sin(d));
12         }
13         printf("\n");
14     }
15     return 0;
```



### 4.3.3 标准库函数

---

- ▶ 2. 实用函数库
- ▶ 实用函数库的头文件为stdlib.h。

### 4.3.3 标准库函数

---

#### ► (1) rand函数

函数原型: `int rand(void);`

函数说明: 返回 `[0, RAND_MAX]` 区间的随机整数, 其中 `RAND_MAX` 是符号常量, 至少为 `32767`。

应用举例: `srand(1); //以1为种子初始化随机数发生器`  
`y=rand(); //得到一个随机整数`

### 4.3.3 标准库函数

---

#### ▶ (2) srand函数

函数原型: `void srand(unsigned int seed);`

函数说明: 以seed作为种子初始化随机数发生器。如果使用相同的seed值调用srand, 则rand函数产生的随机数是重复的。如果没有调用过srand, 则rand函数会自动调用1次srand(1)。

### 4.3.3 标准库函数

---

#### ▶ (3) exit函数

函数原型: `void exit(int status);`

函数说明: 终止程序运行, 且将退出状态status返回给启动本程序的程序。

应用举例: `exit(0);` //程序正常状态终止

### 4.3.3 标准库函数

---



#### 【例4.5】

---

产生 $[0,20)$ 、 $(0,1)$ 区间的10组随机数。

### 4.3.3 标准库函数



#### 例题分析

使用rand函数可以获得随机数。不过每次使用相同的种子调用srand，则产生的随机数总是一样的。如果用系统流逝时间（间隔大于1秒）作为种子，就能产生不同的随机数。为此需要使用time.h中的time(0)调用，它返回从（1970-1-1 0:0:0）起到目前为止所经过的时间，单位为秒。

由于rand产生的随机数在[0,RAND\_MAX]区间，为了得到[a,b)之间的随机整数，可以使用 $(\text{rand}() \% (b-a) + a)$ 式子计算（结果值将含a但不含b）。在a为0的情况下，简写为 $\text{rand}() \% b$ 。用 $\text{rand}() / \text{double}(\text{RAND\_MAX})$ 可以取得0~1之间的随机小数。

### 4.3.3 标准库函数

#### 例4.5

```
1 #include <stdio.h>
2 #include <stdlib.h> //使用实用函数
3 #include <time.h> //使用时间函数
4 int main()
5 {
6     double d;
7     int i,n,seed;
8     seed=time(0); //以系统流逝时间为随机数发生器种子
9     srand((unsigned int)seed);
10    for(i=0;i<10;i++) {
11        n=rand()%20; //产生[0,20)区间的随机整数
12        d=rand()/(double)RAND_MAX; //产生(0,1)区间的随机小数
13        printf("%d %lf\n",n,d);
14    }
15    return 0;
```

### 4.3.4 第三方函数

---

- ▶ 如何调用第三方函数？--以Windows SDK为例子
- ▶ 微软每推出一个Windows，都会同时推出相应的SDK，即软件开发工具包。SDK包含了开发该Windows版本所需的Windows函数和常数定义、API函数说明文档、相关工具和示例。
- ▶ 辅助开发某一类软件的相关文档、范例和工具的集合都可以称为“SDK”。开发Windows平台下的应用程序所使用的SDK只是广义SDK的一个子集。



### 4.3.4 第三方函数

#### ► 在程序中播放Wave音乐：SDK PlaySound函数

函数原型：BOOL PlaySound(LPCSTR pszSound, HMODULE hwnd, DWORD fdwSound);

函数说明：

(1) 参数pszSound指定了要播放声音的字符串，可以是WAVE文件名，或是WAV资源名，或是内存中声音数据指针，或是在系统注册表WIN.INI中定义的系统事件声音。如果该参数为NULL则停止正在播放的声音。

(2) 参数hwnd是应用程序实例句柄，除非pszSound指向一个资源标识符（即fdwSound被定义为SND\_RESOURCE），否则必须设置为NULL。

(3) 参数fdwSound是标志的组合，如下表所示。

(4) 若成功则函数返回TRUE，否则返回FALSE。

应用举例：

```
PlaySound("WindowsXPStart.wav", NULL, SND_SYNC);
```

### 4.3.4 第三方函数

---

#### ▶ 播放Wave音乐程序

```
#include <windows.h>
int main()
{
    PlaySound("WindowsXPStart.wav", NULL, SND_SYNC);
    return 0;
}
```

### 4.3.4 第三方函数

fdwSound标志	含义
SND_APPLICATION	用应用程序指定的关联来播放声音。
SND_ALIAS	pszSound参数指定了注册表或WIN.INI中的系统事件的别名。
SND_ALIAS_ID	pszSound参数指定了预定义的声音标识符。
SND_ASYNC	用异步方式播放声音， PlaySound函数在开始播放后立即返回。
SND_FILENAME	pszSound参数指定了WAVE文件名。
SND_LOOP	重复播放声音， 必须与SND_ASYNC标志一块使用。
SND_MEMORY	播放载入到内存中的声音， 此时pszSound是指向声音数据的指针。

### 4.3.4 第三方函数

fdwSound标志	含义
SND_NODEFAULT	不播放缺省声音，若无此标志，则PlaySound在没找到声音时会播放缺省声音。
SND_NOSTOP	PlaySound不中断原来的声音播出并立即返回FALSE。
SND_NOWAIT	如果驱动程序正忙则函数就不播放声音并立即返回。
SND_PURGE	停止所有与调用任务有关的声音。若参数pszSound为NULL，就停止所有的声音，否则，停止pszSound指定的声音。
SND_RESOURCE	pszSound参数是WAVE资源的标识符，这时要用到hmod参数。
SND_SYNC	同步播放声音，在播放完后PlaySound函数才返回。
SND_SYSTEM	如果是背景窗口,如果这个标志被设置，声音是分配到音频会议系统通知的声音。系统音量控制程序（sndvol）显示音量滑块控制系统通知的声音。设置该标志将下控制音量滑块。如果没有设置该标志，声音是分配到默认的音频会议的应用进程。

**CP 程序设计**