

附录 B C 语言的运算符及其优先级和结合性

优先级	运 算 符	含 义	运算对象的个数	结合方向
1	() [] -> .	圆括号 下标运算符 指向结构体成员运算符 结构体成员运算符		自左至右
2	! ~ ++ -- - (类型) * & sizeof	逻辑非运算符 按位取反运算符 自增运算符 自减运算符 负号运算符 类型转换运算符 指针运算符 地址与运算符 长度运算符	1 (单目运算符)	自右至左
3	* / %	乘法运算符 除法运算符 求余运算符	2 (双目运算符)	自左至右
4	+ -	加法运算符 减法运算符	2 (双目运算符)	自左至右
5	<< >>	左移运算符 右移运算符	2 (双目运算符)	自左至右
6	< <= > >=	关系运算符	2 (双目运算符)	自左至右
7	== !=	等于运算符 不等于运算符	2 (双目运算符)	自左至右
8	&	按位与运算符	2 (双目运算符)	自左至右
9	^	按位异或运算符	2 (双目运算符)	自左至右
10		按位或运算符	2 (双目运算符)	自左至右
11	&&	逻辑与运算符	2 (双目运算符)	自左至右
12		逻辑或运算符	2 (双目运算符)	自左至右
13	? :	条件运算符	3 (三目运算符)	自右至左

续表

优先级	运算符	含义	运算对象的个数	结合方向
14	= + = - = * = /= %= >> = << = & = ^ = =	赋值运算符	2 (双目运算符)	自右至左
15	,	逗号运算符(顺序求值运算符)	2 (双目运算符)	自左至右

使用说明：

- (1) 表中的运算符优先级别由上到下依次递减,逗号运算符优先级最低。
- (2) 同一优先级别的运算符没有优先次序,运算次序由其结合方向来决定。
- (3) 不同的运算符要求有不同的运算对象个数,单目运算符要求运算对象只有一个;双目运算符要求运算对象为两个;三目运算符要求运算对象为三个。
- (4) 对于初学者在使用运算符时,为了避免产生混乱,最好每一种运算式子的两侧加圆括号,以示区分。

附录 D C 库函数

D1. 数学函数

在编程中使用数学函数时,应该在源程序文件前使用以下命令行,以便将所使用的函数定义包含到自己的源程序文件中。

#include<math.h>或#include"math.h"

函数名	函数与形参类型	功 能	返 回 值	说 明
abs	int abs (int x);	求整数 x 的绝对值	计算结果	
acos	double acos (double x);	计算 $\cos^{-1}(x)$ 的值	计算结果	x 应在 $-1 \sim 1$ 内
asin	double asin (double x);	计算 $\sin^{-1}(x)$ 的值	计算结果	x 应在 $-1 \sim 1$ 内
atan	double atan (double x);	计算 $\tan^{-1}(x)$ 的值	计算结果	
atan2	double atan2 (double x, double y);	计算 $\tan^{-1}(x/y)$ 的值	计算结果	
cos	double cos (double x);	计算 $\cos(x)$ 的值	计算结果	x 的 单 位 为 弧度
cosh	double cosh (double x);	计算 x 的双曲余弦 $\cosh(x)$ 的值	计算结果	
exp	double exp (double x);	求 e^x 的值	计算结果	
fabs	double fabs (double x);	求 x 的绝对值	计算结果	
floor	double floor (double x);	求出不大于 x 的最大整数	该整数的双精度实数	
fmod	double fmod (double x, double y);	求整除 x/y 的余数	返回余数的双精度数	
frexp	double frexp (double val, int* eptr);	把双精度数 val 分解成数字部分(尾数)x 和以 2 为底的指数 n, 即 $val = x * 2^n$, n 存放在 eptr 指向的变量中	返回数字部分 x, $0.5 \leq x < 1$	
log	double log (double x);	求 $\ln x$	计算结果	
log10	double log10 (double x);	求 $\lg x$	计算结果	
modf	double modf (double val, double* iptr);	把双精度数 val 分解成数字部分和小数部分, 把整数部分存放到 iptr 指向的单元	val 的小数部分	

续表

函数名	函数与形参类型	功 能	返 回 值	说 明
pow	double pow (double x, double y);	计算 x^y 的值	计算结果	
rand	int rand (void);	产生 -90 ~ 32 767 的随机整数	随机整数	
sin	double sin (double x);	求 $\sin(x)$ 的值	计算结果	x 的单位为弧度
sinh	double sinh (double x);	计算 x 的双曲正弦函数 $\sinh(x)$ 的值	计算结果	
sqrt	double sqrt (double x);	计算 \sqrt{x}	计算结果	$x \geq 0$
tan	double tan (double x);	计算 $\tan(x)$ 的值	计算结果	x 单位为弧度
tanh	double tanh (double x);	计算 x 的双曲正切函数 $\tanh(x)$ 的值	计算结果	

D2. 字符函数和字符串函数

在编程中使用字符串函数时,应该在源程序文件前使用以下命令行,以便将所使用的函数定义包含到自己的源程序文件中。

```
#include<string.h>或#include"string.h"
```

有些 C 编译系统将字符串函数封装成其他文件名的头文件,在使用时要根据所使用的 C 编译系统的版本不同来确定。

函数名	函数原型	功 能	返 回 值	包含文件
isalnum	int isalnum (int ch);	检查 ch 是否是字母(alpha)或数字(numeric)	是字母或数字返回 1;否则返回 0	ctype.h
isalpha	int isalpha (int ch);	检查 ch 是否是字母	是,返回 1;不是则返回 0	ctype.h
isctrl	int isctrl (int ch);	检查 ch 是否控制字符(其 ASCII 码为 0~0x1F)	是,返回 1;不是则返回 0	ctype.h
isdigit	int isdigit (int ch);	检查 ch 是否是数字(0~9)	是,返回 1;不是则返回 0	ctype.h
isgraph	int isgraph (int ch);	检查 ch 是否是可打印字符(其 ASCII 码为 0x21 ~ 0x7E),不包括空格	是,返回 1;不是则返回 0	ctype.h
islower	int islower (int ch);	检查 ch 是否是小写字母(a~z)	是,返回 1;不是则返回 0	ctype.h
isprint	int isprint (int ch);	检查 ch 是否是可打印字符(包括空格),其 ASCII 码为 0x21~0x7E	是,返回 1;不是则返回 0	ctype.h

续表

函数名	函数原型	功 能	返 回 值	包含文件
ispunct	int ispunct (int ch);	检查 ch 是否是标点字符 (不包括空格), 即除字母、数字和空格以外的所有可打印字符	是, 返回 1; 不是则返回 0	ctype.h
isspace	int isspace (int ch);	检查 ch 是否是空格、跳格符(制表符)或换行符	是, 返回 1; 不是则返回 0	ctype.h
isupper	int isupper (int ch);	检查 ch 是否是大写字母 (A~Z)	是, 返回 1; 不是则返回 0	ctype.h
isxdigit	int isxdigit (int ch);	检查 ch 是否是一个十六进制数学字符 (即 0~9, 或 A~F, a~f)	是, 返回 1; 不是则返回 0	ctype.h
strcat	char * strcat (char * str1, char * str2);	把字符串 str2 接到 str1 后面, str1 最后面的 '\0' 被取消	str1	string.h
strchr	char * strchr (char * str, int ch);	找出 str 指向的字符串中第一次出现字符 ch 的位置	返回指向该位置的指针, 如找不到, 则返回空指针	string.h
strcmp	int strcmp (char * str1, char * str2);	比较两个字符串 str1 和 str2	str1 < str2, 返回负数; str1 = str2, 返回 0; str1 > str2, 返回正数	string.h
strcpy	char * strcpy (char * str1, char * str2);	把 str2 指向的字符串复制到 str1 中	返回 str1	string.h
strlen	unsigned int strlen (char * str);	统计字符串 str 中字符的个数 (不包括终止符 '\0')	返回字符个数	string.h
strstr	char * strstr (char * str1, char * str2);	找出 str2 字符串在 str1 中第一次出现的位置 (不包括 str2 的串结束符)	返回该位置的指针。如找不到, 返回空指针	string.h
tolower	int tolower (int ch);	ch 字符转换为小写字母	返回 ch 所代表的字符的小写字母	ctype.h
toupper	int toupper (int ch);	将 ch 字符转换为大写字母	返回与 ch 对应的大写字母	

D3. 输入输出函数

在编程中使用输入输出函数时, 应该在源程序文件前使用以下命令行, 以便将所使用的函数定义包含到其源程序文件中。

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


函数名	函数原型	功 能	返 回 值	说 明
clearerr	void clearerr(FILE * fp);	清除文件指针错误。指示器	无	
close	int close(int fp);	关闭文件	关闭成功返回 0; 不成功返回 -1	非 ANSI 标准
creat	int creat (char * filename, int mode);	以 mode 指定的方式建立文件	成功返回正数; 否则返回 -1	非 ANSI 标准
eof	int eof (int fp);	检查文件是否结束	遇文件结束返回 1; 否则返回 0	非 ANSI 标准
fclose	int fclose (FILE * fp);	关闭 fp 所指的文件, 释放文件缓冲区	有错则返回非 0; 否则返回 0	
feof	int feof (FILE * fp);	检查文件是否结束	遇文件结束符返回非 0; 否则返回 0	
fgetc	int fgetc (FILE * fp);	从 fp 所指定的文件中取得下一个字符	返回所得到的字符。若读入出错, 返回 EOF	
fgets	char * fgets (char * buf, int n, FILE * fp);	从 fp 指向的文件读取一个长度为 (n-1) 的字符串, 存入起始地址为 buf 的空间	返回地址 buf, 若遇文件结束或出错则返回 NULL	
fopen	FILE * fopen (char * filename, char * mode);	以 mode 指定的方式打开名为 filename 的文件	成功, 返回一个文件指针(文件信息区的起始地址); 否则返回 0	
fprintf	int fprintf (FILE * fp, char * format, args, ...);	把 args 的值以 format 指定的格式输出到 fp 所指的文件中	实际输出的字符数	
fputc	int fputc (char ch, FILE * fp);	将字符 ch 输出到 fp 所指的文件中	成功, 则返回该字符; 否则返回非 0	
fputs	int fputs (char * str, FILE * fp);	将 str 指向的字符串输出到 fp 所指定的文件	成功则返回 0; 出错返回非 0	
fread	int fread (char * pt, unsigned size, unsigned n, FILE * fp);	从 fp 所指定文件中读取长度为 size 的 n 个数据项, 存到 pt 所指向的内存区	返回所读的数据项个数, 如遇文件结束或出错返回 0	

函数名	函数原型	功 能	返 回 值	说 明
fscanf	int fscanf (FILE * fp, char format, args, ...);	从 fp 指定的文件中按给定的 format 格式将输入的数据送到 args 所指向的内存单元 (args 是指针)	已输入的数据个数	
fseek	int fseek (FILE * fp, long offset, int base);	将 fp 指向的文件的位置指针移到 base 所指出的位置为基准、以 offset 为位移量的位置	返回当前位置; 否则返回 -1	
ftell	long ftell (FILE * fp);	返回 fp 所指向的文件中读写位置	返回 fp 所指向的文件中读写位置	
fwrite	int fwrite (char * ptr, unsigned size, unsigned n, FILE * fp);	把 ptr 所指向的 n * size 个字节输出到 fp 所指向的文件中	写到 fp 文件中的数据项的个数	
getc	int getc (FILE * fp);	从 fp 所指向的文件中读入一个字符	返回所读的字符, 若文件出错或结束返回 EOF	
getchar	int getchar (void);	从标准输入设备中读取下一个字符	所读字符, 若文件出错或结束返回 -1	
getw	int getw (FILE * fp);	从 fp 所指向的文件读取下一个字(整数)	输入的整数。如文件结束或出错, 返回 -1	非 ANSI 标准函数
open	int open (char * filename, int mode);	以 mode 指定的方式打开已存在的名为 filename 的文件	返回文件号(正数)。如打开失败, 返回 -1	非 ANSI 标准函数
printf	int printf (char * format, args, ...);	按 format 指向的格式字符串所规定的格式, 将输出列表 args 的值输出到标准输出设备	输出字符的个数; 若出错返回负数	format 可以是一个字符串, 或字符数组的起始地址
putc	int putc (int ch, FILE * fp);	把一个字符 ch 输出到 fp 所指的文件中	输出的字符 ch。若出错返回 EOF	
putchar	int putchar (char ch);	把字符 ch 输出到标准输出设备	输出的字符 ch。若出错返回 EOF	

续表

函数名	函数原型	功 能	返 回 值	说 明
puts	int puts (char * str);	把 str 指向的字符串输出到标准输出设备,将'\0'转换为回车换行	返回换行符。若失败返回 EOF	
putw	int putw (int w, FILE * fp);	将一个整数 w (即一个字)写到 fp 指向的文件中	返回输出的整数。若出错返回 EOF	非 ANSI 标准函数
read	int read (int fd, char * buf, unsigned count);	从文件号 fd 所指的文件中读 count 个字节到由 buf 指示的缓冲区中	返回真正读入的字节个数,如遇文件结束返回 0, 出错返回 -1	非 ANSI 标准函数
rename	int rename (char * oldname, char * newname);	把由 oldname 所指的文件名,改为由 newname 所指的文件名	成功返回 0, 出错返回 -1	
rewind	void rewind (FILE * fp);	将 fp 指示的文件中的位置指针置于文件开头位置,并清除文件结束标志和错误标志	无	
scanf	int scanf (char * format, args, ...);	从标准输入设备按 format 指示的格式字符串所规定的格式,输入数据给 args 所指示的单元	读入并赋给 args 的数据个数。遇文件结束返回 EOF; 出错返回 0	args 为指针
write	int write (int fd, char * buf, unsigned count);	从 buf 指示的缓冲区输出 count 个字符到 fd 所标志的文件中	返回实际输出的字节数,如出错返回 -1	非 ANSI 标准函数

D4. 动态存储分配函数

在编程中使用输入输出函数时,应该在源程序文件前使用以下命令行,以便将所使用的函数定义包含到自己的源程序文件中。

```
#include<stdlib.h>或 #include"stdlib.h"
```

函数名	函数和形参类型	功 能	返 回 值
calloc	void * calloc (unsigned n, unsigned size);	分配 n 个数据项的内存连续空间,每个数据项的大小为 size	分配内存单元的起始地址。如不成功,返回 0
free	void free(void * p);	释放 p 所指内存区	无
malloc	void * malloc(unsigned size);	分配 size 字节的存储区	所分配的内存区地址,如内存不够,返回 0
realloc	void * realloc(void * p, unsigned size);	将 p 所指出的已分配内存区的大小改为 size。size 可以比原来分配的空间大或小	返回指向该内存区的指针