附录 常用库函数

Microsoft C为应用程序设计人员提供了大量的库函数，其功能覆盖了程序设计的方方面面，为每个程序员所必须掌握。在本附录中，我们选列了其中较常用的库函数，供读者查询参考。为了节省篇幅，我们只列出了函数原型和简单的功能说明。如果在使用时对某库函数的功能、参数设置和返回值等仍有疑问，可以直接查看相应的头文件或者MSDN中的有关说明。

1．分类库函数，原型均在头文件ctype.h中。这组库函数实际上均为带参数的宏。  
原型： int isalpha（int c）；  
功能： 如果c为字母返回非0值，否则返回0。  
原型： int isdigit（int c）；  
功能： 如果c为数字返回非0值，否则返回0。  
原型： int isalnum（int c）；  
功能： 如果c为字母或数字返回非0值，否则返回0。  
原型： int islower（int c）；  
int isupper（int c）；  
功能： 如果c为小写（大写）字母返回非0值，否则返回0。  
原型： int isprint（int c）；  
功能： 如果c为可打印字符（包括字母、数字、空格和各种符号等）返回非0值，否则返回0。  
原型： int isspace（int c）；  
功能： 如果c为空格、制表符、回车换行符等则返回非0值，否则返回0。  
原型： int tolower（int ch）；  
int toupper（int ch）；  
功能： 将字符转换为小写或大写字符。

2．目录操作函数，用于在程序中处理文件目录系统，原型均在头文件direct.h中。  
原型： int \_chdir（ const char \*dirname ）；  
功能： 改变当前目录。  
原型： int \_chdrive（ int drive ）；  
功能： 改变当前驱动器。  
原型： char \*\_getcwd（ char \*buffer, int maxlen ）；  
功能： 读取当前目录。  
原型： char \*\_getdcwd（ int drive, char \*buffer, int maxlen ）；  
功能： 从指定驱动器的读取当前目录。  
原型： int \_getdrive（ void ）；  
功能： 读取当前磁盘驱动器。  
原型： int \_mkdir（ const char \*dirname ）；  
功能： 建立目录。  
原型： int \_rmdir（ const char \*dirname ）；  
功能： 删除目录。  
原型： void \_searchenv（ const char \*filename, const char \*varname, char \*pathname ）；  
功能： 按指定路径查找给出的文件。

3．进程和环境控制库函数，原型均在头文件process.h或stdlib.h中。  
原型： void abort（void）；  
功能： 异常终止一个进程。  
原型： void exit（int status）；  
功能： 中止程序。  
原型： int system（char \*command）；  
功能： 在不退出程序的情况下直接启动DOS的命令行。可以使用此函数运行其它程序，然后再返回原来的应用程序。

4．类型转换库函数，除注明者外，原型均在头文件stdlib.h中。  
原型： int atoi（char \*s）；  
long atol（char \*s）；  
double atof（char \*s）；  
功能： 将字符串分别转换为整型数、长整型数和双精度型数。atof的原型在头文件math.h中。  
原型： char \*\_itoa（int value，char \*string，int radix）；  
char \*\_ltoa（long value, char \*string，int radix）；  
功能： 整型或长整型数值转换为字符串。

5．输入/输出库函数，除注明者外，原型均在头文件stdio.h中。  
原型： int getchar（ void ）；  
功能： 从标准输入设备（键盘，下同）读一个字符。  
原型： int getch（void）；  
int getche（void）；  
功能： 从标准输入设备（键盘,下同）读一个字符。原型在头文件conio.h中。  
原型： int \_kbhit（ void ）；  
功能： 检查当前按键是否有效。原型在头文件conio.h中。  
原型： int putchar（ int c ）；  
功能： 向标准输出设备（显示屏，下同）写一个字符。  
原型： int \_putch（ int c ）；  
功能： 向控制台写一个字符。  
原型： int printf（ const char \*format [, argument]... ）；  
功能： 向标准输出设备进行格式化输出。  
原型： int scanf（ const char \*format [,argument]... ）；  
功能： 从标准输入设备进行格式化输入。  
原型： char \*gets（ char \*buffer ）；  
功能： 从标准输人设备输入字符串。  
原型： int puts（ const char \*string ）；  
功能： 向标准输出设备写字符串。  
原型： int \_ungetch（ int c ）；  
功能： 将一个字符退回键盘缓冲区，下次用getch（）等函数即可读此字符。  
原型： FILE \*fopen（ const char \*filename, const char \*mode ）；  
int fclose（ FILE \*stream ）;  
功能： 打开和关闭流式文件。  
原型： int fgetc（ FILE \*stream ）；  
int getc（ FILE \*stream ）；  
int fputc（ int c, FILE \*stream ）；  
int putc（ int c, FILE \*stream ）；  
功能： 从文件中读一个字符，或向文件中写一个字符。  
原型： char \*fgets（ char \*string, int n, FILE \*stream ）；  
int fputs（ const char \*string, FILE \*stream ）；  
功能： 文件字符串读写。  
原型： int fprintf（ FILE \*stream, const char \*format [, argument ]...）；  
int fscanf（ FILE \*stream, const char \*format [, argument ]... ）；  
功能： 文件格式化读写。  
原型： size\_t fread（ void \*buffer, size\_t size, size\_t count, FILE \*stream ）；  
size\_t fwrite（ const void \*buffer, size\_t size, size\_t count, FILE \*stream ）；  
功能： 文件成块读写。  
原型： int fseek（ FILE \*stream, long offset, int origin ）；  
long ftell（ FILE \*stream ）；  
功能： 文件指针定位。  
原型： int feof（ FILE \*stream ）；  
功能： 测试文件是否结束。  
原型： int ungetc（ int c, FILE \*stream ）；  
功能： 将一个字符退回文件缓冲区中，下次用fgetc（）等函数即可读此字符。  
原型： int rename（ const char \*oldname, const char \*newname ）；  
功能： 修改文件名。  
原型： int \_unlink（ const char \*filename ）；  
功能： 删除文件。原型在头文件stdio.h和io.h中。

6．存储分配库函数，原型均在头文件malloc.h中。  
原型： void \*malloc（ size\_t size ）；  
功能： 按字节申请存储空间。  
原型： void \*calloc（ size\_t num, size\_t size ）；  
功能： 成组申请存储空间。  
原型： void \*realloc（ void \*memblock, size\_t size ）；  
功能： 修改已动态分配的存储块大小。  
原型： void free（ void \*memblock ）；  
功能： 释放动态申请的存储块。

7．内存操作函数，除特别说明者外，原型均在头文件memory.h和string.h中。  
原型： void \*memset（ void \*dest, int c, size\_t count ） ；  
功能： 将一段内存均置为给定的内容。  
原型： int memcmp（ const void \*buf1, const void \*buf2, size\_t count ）；  
int \_memicmp（ const void \*buf1, const void \*buf2, unsigned int count ）；  
功能： 比较两段存储空间的内容。\_memicmp（）不区分大、小写字母。  
原型： void \*memchr（ const void \*buf, int c, size\_t count ）；  
功能： 在指定内存单元中查找字符。  
原型： void \*memcpy（ void \*dest, const void \*src, size\_t count ）；  
void \*memmove（ void \*dest, const void \*src, size\_t count ）；  
功能： 复制一段给定的内存单元内容。memcpy（）与memmove（）的复制方向不同，这在源空间和目标空间有重叠时特别重要。

8．字符串操作库函数，原型均在头文件string.h中。这组库函数甚多，这里只选列了其中较常用的几个。  
原型： char \*\_strset（ char \*string, int c ）；  
char \*\_strnset（ char \*string, int c, size\_t count ）；  
功能： 将字符串中的字符设置为给定值。  
原型： char \*strcpy（ char \*strDestination, const char \*strSource ） ；  
char \*strncpy（ char \*strDest, const char \*strSource, size\_t count ）；  
功能： 字符串拷贝。  
原型： char \*strcat（ char \*strDest, const char \*strSource ）；  
char \*strncat（ char \*strDest, const char \*strSource, size\_t count ）；  
功能： 将字符串strSource 添加在字符串strDest之后。  
原型： char \*\_strlwr（ char \*string ）；  
char \*\_strupr（ char \*string ）；  
功能： 将字符串中的字母字符均转换为小写／大写字母。  
原型： char \*\_strrev（ char \*string ）；  
功能： 反转字符串的内容。  
原型： char \*strchr（ const char \*string, int c ）；  
功能： 在字符串string中查找指定字符c  
原型： char \*strstr（ const char \*string, const char \*strCharSet ）；  
功能： 在字符串中查找指定子字符串。  
原型： int strcmp（ const char \*string1, const char \*string2 ）；  
int \_stricmp（ const char \*string1, const char \*string2 ）；  
int strncmp（ const char \*string1, const char \*string2, size\_t count ）；  
int \_strnicmp（ const char \*string1, const char \*string2, size\_t count ）；  
功能： 比较两个字符串。stricmp（）在比较时不区分大小写字母；strncmp（）仅比较两字符串的前maxlen位；strnicmp（）仅比较两字符串的前maxlen位，且比较时不区分大小写字母。如果两字符串相同，则返回0否则返回非0值。  
原型： size\_t strlen（ const char \*string ）；  
功能： 求字符串长度。

9．数学运算库函数，原型均在头文件math.h。  
原型： int abs（ int n ）;  
double fabs（double x）;  
long labs（long x）；  
功能： 求绝对值。  
原型： double \_cabs（ struct \_complex z ）；  
功能： 求复数的模。  
原型： double sin（double）；  
double cos（double x）；  
double tan（double x）；  
double asin（double x）；  
double acos（double x）；  
double atan（double x）；  
double atan2（double y， double x）；  
功能： 求三角函数和反三角函数。  
原型： double sinh（double）；  
double cosh（double x）；  
double tanh（double x）；  
功能： 求双曲函数。  
原型： double exp（double）；  
double log（double x）；  
double log10（double x）；  
double pow（double x， double y）；  
double sqrt（double x）；  
功能： 求指数、对数、幂函数和平方根。  
原型： int max（int a，int b）；  
int min（int a，int b）；  
功能： 求两数中的较大者或较小者。这两个库函数均为定义于头文件stdlib.h中的带参数的宏。  
原型： double \_hypot（ double x, double y ）；  
功能： 计算直角三角形斜边的长。

10．时间函数，原型均在头文件 time.h 中。  
原型： time\_t time ( time\_t \*timer );  
功能： 取当前系统时间。其中类型 time\_t（相当于long int）存放自1970年1月1日午夜起流逝的秒数。  
原型： char \*ctime ( const time\_t \*timer );  
功能： 将以 time\_t 格式存放的时间转换为相应的字符串，结果为一静态字符串的地址，其格式形如：“Wed Jan 02 02:03:55 1980\n”。  
原型： double difftime ( time\_t timer1, time\_t timer0 );  
功能： 求从timer0到timer1之间以秒为单位的时间间隔。  
原型： struct tm \*localtime ( const time\_t \*timer );  
struct tm \*gmtime( const time\_t \*timer );  
功能： 将以 time\_t 类型表示的时间值转换为以 struct tm 格式表示的时间。struct tm 的格式为：  
struct tm {  
int tm\_sec; /\* seconds after the minute - [0,59] \*/  
int tm\_min; /\* minutes after the hour - [0,59] \*/  
int tm\_hour; /\* hours since midnight - [0,23] \*/  
int tm\_mday; /\* day of the month - [1,31] \*/  
int tm\_mon; /\* months since January - [0,11] \*/  
int tm\_year; /\* years since 1900 \*/  
int tm\_wday; /\* days since Sunday - [0,6] \*/  
int tm\_yday; /\* days since January 1 - [0,365] \*/  
int tm\_isdst; /\* daylight savings time flag \*/  
};  
原型： time\_t mktime ( struct tm \*timeptr );  
功能： 将struct tm 格式的时间转换为 time\_t 格式的时间。  
原型： char \*asctime( const struct tm \*timeptr );  
功能： 将 struct tm 格式的时间转换为字符串（参看 ctime（）函数）。  
  
11．随机数发生器库函数，原型均在头文件stdlib.h中。  
原型： int rand（ void ）；  
功能： 产生一个范围在0~RAND\_MAX（0x7fff〕之间的随机数。  
原型： void srand（ unsigned int seed ）；  
功能： 根据给定的初值初始化随机发生器。例如：  
time\_t timer;  
time (&timer);  
srand ((unsigned int)timer);  
用当前系统时间初始化随机数发生器，用以得出不重复的随机数序列。