附录 1　C-Lib 中的函数集

表 F1　字符函数集一览

序号	调用方式	功能详述
1	int isalnum(int c)	若参数 c 是一个字母(‘A’—‘Z’ 或‘a’—‘z’)或是一个数字(‘0’—‘9’)，则函数返回非零值，否则返回零
2	int isalpha(int c)	若参数 c 为字母表中的字母(‘A’—‘Z’或‘a’—‘z’)，则函数返回非零值，否则返回零
3	int isascii(int c)	若参数 c 为 ASCII 字符，即 c 的低字节在 0～127(0x00～0x7F)之间，则函数返回非零值，否则返回零
4	int iscntrl(int c)	若参数 c 为控制符，即 c 为删除（delete）控制符（0x7F）或一般控制符(0x00～0x1F)，则函数返回非零值，否则返回零
5	int isdigit(int c)	若参数 c 为十进制数字(0～9)，则函数返回非零值，否则返回零
6	int islower(int c)	若参数 c 为小写字母(‘a’～‘z’)，则函数返回非零值，否则返回零
7	int isprint(int c)	若参数 c 为可打印字符(0x20～0x7E)，则函数返回非零值，否则返回零
	int isgraph(int c)	与 isprint 函数类似，只是不包含空格符(0x20)
8	int ispunct(int c)	若参数 c 为可打印的标点符号，即(isgraph(c) && !isalnum(c))为非零值，则函数返回非零值，否则返回零
9	int isspace(int c)	若参数 c 为空格、水平或垂直制表符、回车符、换行符及进纸符等(0x09--0x0D，0x20)，则函数返回非零值，否则返回零
10	int isupper(int c)	若参数 c 为大写字母(‘A’～‘Z’)，则函数返回非零值，否则返回零
11	int isxdigit(int <[c]>)	若参数 c 为十六进制数字(‘0’～‘9’，‘a’～‘f’，或‘A’～‘F’)，则函数返回非零值，否则返回零
12	int toascii(int c)	将整型参变量 c 转换成范围为 0～127 的 ASCII 码
13	int tolower(int c)	将整型参变量 c 转换成小写字母值（‘a’～‘f’）
	int _tolower(int<[c]>)	与 tolower 函数功能相似，但本函数只能用于已知参数是大写字母（‘A’～‘Z’）的情况，否则函数的返回值不定
14	int toupper(int c)	将整型参变量 c 转换成大写字母值（‘A’～‘Z’）
	int _toupper(int<[c]>)	与 toupper 函数功能相似，但本函数只能用于已知参数是小写字母（‘a’～‘f’）的情况，否则函数的返回值不定

表 F2　字符串和内存函数集一览

序号	调用方式	功能详述
1	int bcmp(const char *s1, const char *s2, int n)	比较由参数 s1 和 s2 指向的两数组的前 n 个字符，若 s1 指向的数组大于、等于或小于 s2 指向的数组，函数分别返回一个大于、等于或小于 0 的整数
2	void bcopy(const char *in, char *out, int n)	将参数 in 指向的存储区的 n 个字节复制到参数 out 指向的存储区
3	void bzero(char *b, int length)	将指针为参数 b，长度为参数 length 的存储区内数据初始化为零
4	void* charpack(void *dst, char *s)	将由参数 s 所指的以 16 位数表示的字符串压缩转换成由参数 dst 所指的以 8 位数表示的字符串，并返回指针 dst 值。被压缩字符串中的空字符会结束转换

续表 F2

序号	调用方式	功能详述
5	void* charunpack(void*dst,void*s)	将由参数 s 所指的以 8 位数表示的字符串转换成由参数 dst 所指的以 16 位数表示的字符串，并返回指针 dst 值。解压缩字符串中的空字符会结束转换
6	char* index(const char *string,int c)	在参数 string 指向的字符串中（包含结束符）查找已转换成字符的参数 c，并返回查到 c 处的指针。若未查找到 c，则返回空指针
7	void* memccpy (void *out, const void *in, int c, int n)	将参数 in 所指内存内容拷贝到参数 out 所指内存中去，当拷贝了 n 个字节后或者第一次遇到的参数 c 被拷贝后就停止拷贝操作。若拷贝了 c，则返回指向 out 中紧跟 c 后字符的指针，否则返回空指针
8	void* memchr(const void *src, int c, int length)	在由参数 src 所指的存储区中前 length 个字节内搜索参数 c，返回一个指向在 src 中最先遇到的 c 的指针，若未找到 c 则返回空指针
9	int memcmp(const void *s1, const void *s2, int n)	比较参数 s1、s2 指向的数组的前 n 个字符。若 s1 指向的数组大于、等于或小于由 s2 指向的数组，函数分别返回一个大于、等于或小于零的整数。若两数组比较区域重叠，则函数的行为非法
10	void* memcpy(void *out, const void *in, int n)	从参数 in 所指的数组中拷贝 n 个字符到参数 out 所指的数组中去，并返回一个指向 out 的指针。若两数组区域重叠，函数的行为非法
11	void* memmove(void *dst,const void *src,unsigned length)	从 src 所指的数组中拷贝 length 个字符到 dst 所指的数组中去，并返回一个指向 dst 的指针。数组区域重叠不会影响内容正确地拷贝
14	char *rindex(const char *string, int c)	在参数 string 所指的字符串中(包括结束符)，查找最后一次出现的已转换为字符型的参数 c，并返回查到 c 处的指针，如未找到 c，则返回一个空指针。本函数与函数 strrchr 功能相同
15	void setmem(char *dst,int length,char c)	将参数 dst 所指的数组中 length 个字节的块都设置为字符参数 c
16	char *stpcpy(char *dst, const char *src)	将参数 src 指向的数组中字符串拷贝到参数 dst 指向的数组中,直到遇到结束符为止，并返回指针 dst+strlen(src)之值
17	int strcasecmp(const char *a,const char*b)	将参数 a、b 所指的两个字符串进行比较，并识别大小写。若按字典顺序，a 排在 b 后（都转换为大写），函数返回一个正数；若 a 排在 b 前，函数返回一个负数；若两个字符串匹配，函数返回零
18	char *strcat(char *dst, const char *src)	把参数 src 指向的字符串(包含结束符)连到参数 dst 指向的字符串尾部，src 第一个字符覆盖 dst 的结束符，并返回合并后字符串指针
19	char * strchr(const char *string,int c)	返回参数 string 指向的字符串中（包含结束符）首次出现的字符参数 c 的位置指针；若未发现与 c 匹配的字符，则返回空指针
20	int strcmp(const char *a, const char *b)	对参数 a、b 所指的字符串进行比较。按字典顺序，若 a 排在 b 后，函数返回一个正数；若 a 排在 b 前，函数返回一个负数；若两个字符串匹配，则函数返回零
21	int strcoll(const char *stra, const char * strb)	对参数 stra、strb 所指的字符串进行比较。据 stra 字符串大于、小于以及等于 strb 字符串，函数将相应返回一个正数、负数及零
22	char *strcpy(char *dst, const char *src)	将参数 src 指向的字符串（包含结束符）拷贝到参数 dst 指向的数组中去，并返回 dst 的指针。若两数组区域重叠，则函数行为非法
23	int strcspn(const char *s1, const char *s2)	返回参数 s1 所指字符串的初始子串的长度，该子串中的任一字符都不包含于参数 s2 所指的字符串中（结束符除外）

续表 F2

序号	调用方式	功能详述
24	int strlen(const char *str)	计算并返回参数 str 所指字符串的长度。方法是计数字符个数直至结束符为止（结束符不计在内）
25	char *strlwr(char *a)	把参数 a 指向的字符串中每个字符都变为小写字母，并返回指针 a
26	int strncasecmp(const char *a, const char *b, int length)	对参数 a、b 指向的两个字符串中 length 个字符进行比较，并识别其大小写。按字典顺序，若 a 排在 b 后（都转换为大写），函数返回一个正数；若 a 排在 b 前，函数返回一个负数；若两个字符串匹配，则函数返回零
27	char *strncat(char *dst,const char *src, int length)	将参数 src 指向的字符串（包含结束符）中前 length 个字符连接到参数 dst 指向的字符串之尾部，src 中第一个字符覆盖 dst 的结束符。函数返回 dst 指针值。
28	int strncmp(const char *a, const char *b, int length)	比较参数 a、b 指向的两个字符串中前 length 个字符。按字典顺序，若 a 排在 b 后（都转换为大写），函数返回一个正数；若 a 排在 b 前，函数返回一个负数；若两个字符串匹配，函数返回零
29	char *strncpy(char *dst, const char *src, intlength)	将参数 src 指向的字符串（包括结束符）中前 length 个字符拷贝到参数 dst 指向的数组中去。若 src 数组中少于 length 个字符，会在 dst 数组中添加空字符至凑够 length 个字符。函数返回 dst 指针值
30	char *strnset(char *dst, char c, unsigned n)	将参数 dst 指向的字符串中前 n 个字符设置为字符参数 c 的值
31	char *strpbrk(const char *s1, const char *s2)	在参数 s1 指向的字符串中查找与参数 s2 指向的字符串中任何一个字符相匹配的第一个字符（空字符不包含在内），并返回其位置指针。若没有匹配字符，则返回空指针
32	char *strrchr(const char *string,int c)	在参数 string 指向的字符串中（包含结束符）查找最后一次出现字符参数 c，并返回其位置指针；若未找到 c，则返回空指针
33	char *strrev(char *s)	将参数 s 指向的字符串中所有字符顺序都颠倒过来（结束符除外），并返回指向颠倒顺序后的字符串指针
34	char *strset(char *s, char c)	将参数 s 所指字符串中所有字符都设置成字符参数 c，并返回指针 s
35	int strspn(const char *s1, const char *s2)	在参数 s1 所指字符串中查找第一个不属于参数 s2 所指字符串中字符的位置（结束符除外），计算并返回从起始到此位置的长度值
36	char *strstr(const char *s1, const char *s2)	在参数 s1 所指字符串中查找第一次遇到参数 s2 所指字符串（其结束符除外），并返回其位置指针；若未找到相匹配的字符串，返回空指针；若 s2 指向的字符串长度为零，则返回指针 s1
37	char *strtok(char *source, const char, *delimiters)	返回参数 source 所指字符串中指向下一个由参数 delimiters 指定的字符或字符串的分隔符的指针，若无分隔符则返回一个空指针。函数实际上修改了由 source 指向的字符串。每找到一个分隔符后，一空字符就被放到分隔符处。函数用此方法连续查遍该字符串
38	char *strupr(char *a)	将参数 a 指向的字符串中所有字符都变为大写字母，并返回指针 a
39	int strxfrm(char *s1, const char *s2, int n)	对参数 s2 指向的字符串进行转换后前 n 个字符（包含结束符）置入参数 s1 指向的数组中；若 n 为 0，则 s1 为空指针。拷贝若在重叠的区域内会使函数结果不定。字符串转换如下：若用于两个转换字符串的函数 strcmp 与用于两个相同的原字符串的函数 strcoll 的结果相对应，函数会分别返回一个正数、零及负数
40	void swab(char *from, char *to, unsigned n)	从参数 from 指向的字符串中拷贝 n 个字符到参数 to 指向的字符串中，并交换相邻的偶、奇数字节

表 F3　数学函数集一览

序号	调用方式	功能详述
1	float acosf(float x)	返回参变量 x 的反余弦值，以弧度表示；x 的定义域为[-1，1]
2	float acoshf(float x)	返回参变量 x 的反双曲余弦值
3	float asinf(float x)	返回参变量 x 的反正弦值，以弧度表示；x 的定义域为[-1，1]
4	float asinhf(float x)	返回参变量 x 的反双曲正弦值
5	float atanf(float x)	返回参变量 x 的反正切值，以弧度表示
6	float atan2f(float y，float x)	返回参变量 y/x 的反正切值
7	float atanhf(float <[x]>)	返回参变量 x 的反双曲正切值
8	float cabs(struct complex z)	返回复数参变量 z 的绝对值
9	float cbrtf(float x)	返回参变量 x 的立方根值
10	float ceilf(float x)	返回不小于参变量 x 的最小整数
11	float copysignf (float x, float y)	构造一个数，其值为参量 x 的绝对值，其符号为参量 y 的符号
12	float cosf(float x)	返回参变量 x 的余弦值，以弧度表示
13	float coshf(float x)	返回参变量 x 的双曲余弦值
14	float dremf (float x, float y)	返回参变量运算 x/y 的余数值
15	float erff(float x)	估算落在参数 x 标准平均误差范围内的概率（假设为正态分布）统计值
	float erfcf(float x)	直接计算函数 erff 的互补概率（1-erff(x)）。用此函数可以避免通过计算 1-erff(x)造成的精度损失
16	float expf(float x)	计算并返回参变量 x 的指数值，即 e^x；e 约为 2.71828
17	float expm1f(float x)	计算并返回（$e^x - 1$）；参变量 x 值即使很小，亦能保证精度。但若使用 expf 函数计算（$e^x - 1$），则会丢失有效位
18	float fabsf(float x)	返回参变量 x 的绝对值
19	int finitef(float x)	若参变量 x 为有限值，返回非零值，否则返回零
20	float floorf(float x)	返回不大于参变量 x 的最大整数
21	float fmodf(float x, float y)	返回浮点数单精度型参变量运算 x/y 的余数
22	float frexpf(float val, int *exp)	把参量 val 分解成一个 0.5~1 范围内的尾数和一个整型指数，即 val=尾数*2^{exp}；其中尾数由函数返回，指数存储在参量 exp 中
23	float gammaf(float x)	计算参变量 x 的 gamma 函数的自然对数
	float lgammaf(float x)	是 gammaf 函数的别名
	float gammaf_r(float x, int *signgamp)	计算参变量 x 的 gamma 函数的自然对数，并将 gamma 函数的符号存储在参量 signgamp 中
	float lgammaf_r(float x, int *signgamp)	同上
24	float hypotf(float x, float y)	返回据给定直角三角形的两个直角边算出其斜边的长度值
25	int ilogbf(float val)	所有非零数都可表示为 m*2^p。若参变量 val 的定义域在 0～INF[1]范围内，函数返回 p；若 val 定义为零，返回-INT_MAX；定义为 INF 或超出定义域，则函数返回 INT_MAX
26	float infinityf(void)	返回 INF 值
27	int isinff(float x)	若参变量 x 为 INF 值，返回非零值，否则返回零
28	int isnanf(float arg)	若参数变量为 NAN[2]值，返回非零值，否则返回零

续表 F3

序号	调用方式	功能详述
29	float j0f(float x)	求解微分方程的第一类贝塞尔函数的零阶特例函数
	float j1f(float x)	求解微分方程的第一类贝塞尔函数的一阶特例函数
	float jnf(int n, float x)	求解微分方程的第一类 n 阶贝塞尔函数
	float y0f(float x)	求解微分方程的第二类贝塞尔函数的零阶特例函数
	float y1f(float x)	求解微分方程的第二类贝塞尔函数的一阶特例函数
	float ynf(int n, float x)	求解微分方程的第二类 n 阶贝塞尔函数
30	float ldexpf(float val, int exp)	计算并返回 val*2^{exp} 的值，若计算发生溢出，则返回 HUGE_VAL
31	float logf(float x)	返回参变量 x 的自然对数值；x 的定义域为（0，INF）
32	float log10f(float x)	返回参变量 x 以 10 为底的对数值；x 的定义域为（0，INF）
33	float log1pf(float x)	返回（1+x）的自然对数值；参量 x 很小时用此函数运算精度高
34	int logbf (float val)	同 ilogbf( )函数
35	int matherr(struct exception *err)	函数用于常见的数学错误处理，用类型为 exception 的参数调用,其结构为：struct exception { int type;char *name; double arg1, arg2, retval;int err;};
36	float modff(float val, float *ipart)	将参变量 val 分解成整数部分和小数部分；其中小数由函数返回，整数则存储在参变量 ipart 中
37	float nanf(void)	返回 NAN 值
38	float nextafterf(float val, float dir)	返回参量 val（IEEE 格式）向参量 dir 方向变化的下一个数值
39	float polyf(float x, int n,float c[ ])	返回计算 x^n 之系数为 c[0]～c[n]的运算多项式的值
40	float powf(float x, float y)	返回计算以参变量 x 为底的 y 次幂，即 x^y 之值
41	float remainderf(float x, float y)	同 dremf()函数
42	float rintf(float x)	返回将参变量 x 经四舍五入处理后的整数值
43	float scalbf(float x, float n)	返回 x*2^n 之运算值，n 为单精度型数
44	float scalbnf(float x, int n)	返回 x*2^n 之运算值，n 为整型数
45	float significandf(float x)	返回函数 scalbf(x，(float) -ilogb(x))调用的结果值
46	float sinf(float x)	返回参变量 x 的正弦值，x 以弧度表示
47	float sinhf(float x)	返回参变量 x 的双曲正弦值
48	float sqrtf(float x)	返回参变量 x 的平方根值；x 的定义域为[0，INF]
49	float tanf(float x)	返回参变量 x 的正切值，x 以弧度表示
50	float tanhf(float x)	返回参变量 x 的双曲正切值；x 以弧度表示

注：

INF 值即为单精度型数的下限值。

NAN 值即为超出参变量定义域范围之值。

表 F4 标准库函数集

序号	调用方式	功能详述
1	void abort(void) ;	程序检测到一种无法处理的异常情况时终止程序的运行
2	int abs(int x);	返回计算整型参数 x 的绝对值
3	void assert(int <[expression]>);	用于在程序中嵌入调试诊断信息的宏。若程序正常运行，表达式参数[expression]为非零值；而当程序运行出现异常，则[expression]为零值，可调用 abort()函数以终止程序的运行
4	float atoff(char *s);	返回将参数 s 所指字符串的起始部分转换的单精度数；若转换未成功（包括溢出），返回‘0.0’；若转换值超出了其可代表数的范围，则会返回‘-HUGE_VAL’或‘HUGE_VAL’
5	int atoi(char *s);	返回将参数 s 所指的字符串转换成的整型数，转换未成则返回零
6	long atol(const char *s);	返回将参数 s 所指的字符串转换成的长整型数；转换未成则返回零
7	void *bsearch(const void *key, const void *base, size_t nmemb, size_t size,int (*compar)(const void *, const void *))	在参数 base 所指的排序数组中执行二元搜索，并返回指向与 key 所指关键字相匹配的第一个元素的指针；若数组未含关键字，则返回空指针。数组中元素数目由参数 nmemb 指定，且每个元素的大小（以字节表示）由参数 size 给定。数据类型 size_t 在 stdlib.h 中被定义为 unsigned int
8	void *calloc(size_t <[n]>, size_t <[s]>);	返回为具有 n 个长度为 s 的数据的数组分配内存区域第一字节的指针；若无足够的内存可分配，则返回空指针
9	div_t div(int n, int d)	将两整型数相除的商和余数返回在结构型参数 div_t 中
10	char*ecvtf(float val,int chars, int *decpt, int *sgn);	将单精度浮点型参数 val 转换成长度为参数 chars 的字符串，并返回指向该字符串的指针。参数 decpt 指向小数点的位置，而参数 sgn 则指向符号变量。fcvtf()函数的参数 decimals 与 chars 不同，它指定的是小数点后的数值转换成字符串的长度
	char*fcvtf(float val,intdecimals,int*decpt,int*sgn);	
	char *gcvtf(float val, int precision, char *buf);	将单精度浮点型参数 val 转换成长度为参数 precision 的字符串，参数 buf 作为指向该字符串数组的指针而被返回
11	void exit(int <[code]>) ;	使得程序立即正常终止运行。状态参数<[code]>被传递到调用过程，若其为零，则表明程序正常终止；若其为非零值，则表明存在执行错误。
12	long labs(long x);	返回计算长整型参数 x 的绝对值
13	ldiv_t ldiv(long n, long d);	两长整型参数 n、d 相除，商和余数返回在结构型参数 div_t
14	void *malloc(size_t <[nbytes]>);	返回申请分配大小（以字节表示）为参数<[nbytes]>的内存区域首字节的指针，若申请未成功则返回空指针。
	void free(void *<[aptr]>);	释放由参数<[aptr]>指向的内存区域，并将它返回给堆
15	void qsort(void *base, size_t nmemb, size_t size, int (*compar) (const void *, const void *));	对参数<[base]>指向的数组中<[nmemb]>个元素进行分类，且每个元素的大小由参数<[size]>定义。参数 compar 用于指向一个比较函数，其中每一参数都指向<[base]>数组中某一元素；据第一参数大于、等于及小于第二参数，则比较函数分别会返回一个正数、零及负数
16	int rand(void);	返回伪随机数序列中在 0～RAND_MAX（包括 RAND_MAX）之间的一个整数。
17	void *realloc(void *<[aptr]>, size_t <[nbytes]>);	将参数<[aptr]>指向的已分配的内存大小变成由参数<[nbytes]>确定新的大小的内存块，并返回指向新块首字节的指针。若堆中分配不出<[nbytes]>个字节，则函数返回空指针
18	void srand(unsigned int seed);	建立由 rand()函数所产生的伪随机数序列中数值的起始点，它允许多个程序用不同的伪随机数序列运行

续表 F4

序号	调用方式	功能详述
19	char *strdup(_CONST char *str);	按参数 str 所指字符串的长度开出内存区，且将字符串内容拷贝到该存储区域并返回指向该区域首字节的指针
20	float strtodf(const char *str, char **endptr);	返回将参数 str 所指的以数值形式表示的字符串转换成一个单精度型数，参数 *endptr 指向转换字符串的结束符（null）。若转换未成功或转换值溢出，函数返回零；若转换值超出其所能代表数的范围，则函数分别会返回±HUGE_VAL
21	long strtol(const char *s, char **ptr, int base);	返回将参数 s 所指的以数值形式表示的字符串转换成的一个长整型数，数值的进制由参数 base 确定。若转换未成功，函数返回零；若转换值上、下溢出，则函数分别会返回 LONG_MAX 及 LONG_MIN
22	unsigned long strtoul(const char *s, char **ptr, int base);	功能与 strtol()函数类似。不同之处在于本函数将字符串转换成一无符号长整型数
23	int system(char *s);	从一正在执行的 C 程序中执行系统的命令。参数 s 指向该命令字符串。若函数调用成功，返回零；否则返回非零值。

注：表中带阴影部分为虚函数。

表 F5 I/O 函数集一览

序号	调用方式	功能详述
1	int printf(const char *format,...)	按参数 format 指定的格式，将其后参量表中列出的参数写到流文件[注]中去。format 可指定说明符的格式参见表 8.7

注：

程序中若需调用 printf（）函数，应使μ'nSP™ IDE 运行在 Simulator 方式下并进行如下操作：选择 Project 菜单的 setting 选项，进入 Device 属性页，在 Device Set 中会有一个缺省的 I/O 口地址 0x7016，选择 Output 单选按钮后，Sound 复选框会被激活，但不要选择 Sound；在 Output File 文本框中输入流文件名即可（详见第六章中「项目的设置」内容）。

表 F6 针对错误号 errnum 的错误信息

errnum	错误信息串	含义解释
E2BIG	Arg list too long	参数表太长
EACCES	Permission denied	不允许
EADDRINUSE	Address already in use	地址已被占用
EADV	Advertise error	警告错误
EAFNOSUPPORT	Address family not supported by protocol family	地址体系超出规定范围
EAGAIN	No more processes	没有更多的步骤
EALREADY	Socket already connected	接口已经连接
EBADF	Bad file number	错误的文件号
EBADMSG	Bad message	错误的信息
EBUSY	Device or resource busy	设备或资源正被使用
ECHILD	No children	无子系统
ECOMM	Communication error	通讯错误
ECONNABORTED	Software caused connection abort	软件错误引起连接失败
ECONNREFUSED	Connection refused	连接未成功
EDEADLK	Deadlock	死锁
EDESTADDRREQ	Destination address required	未给目标地址
EEXIST	File exists	文件已存在

续表 F6

errnum	错误信息串	含义解释
EDOM	Math argument	计算函数参数的域错误
EFAULT	Bad address	错误的地址
EFBIG	File too large	文件太长
EHOSTDOWN	Host is down	主机故障
EHOSTUNREACH	Host is unreachable	主机功能达不到
EIDRM	Identifier removed	标识符丢失
EINPROGRESS	Connection already in progress	连接已在处理中
EINTR	Interrupted system call	中断系统调用
EINVAL	Invalid argument	非法参数
EIO	I/O error	输入/输出错误
EISCONN	Socket is already connected	插口已被连接
EISDIR	Is a directory	路径错误
ELIBACC	Cannot access a needed shared library	要求共享的库不能被访问
ELIBBAD	Accessing a corrupted shared library	要访问的共享库已被破坏
ELIBEXEC	Cannot exec a shared library directly	不可直接执行一个共享库
ELIBMAX	Attempting to link in more shared libraries than system limit	链接的共享库已超出限度
ELIBSCN	<<.lib>> section in a.out corrupted	在*.out 文件中的<.lib>损坏
EMFILE	Too many open files	要打开的文件太多
EMLINK	Too many links	要链接的模块太多
EMSGSIZE	Message too long	信息太长
EMULTIHOP	Multihop attempted	非法的多重接收
ENAMETOOLONG	File or path name too long	文件或路径名太长
ENETDOWN	Network interface not configured	网络接口未配置
ENETUNREACH	Network is unreachable	网络功能达不到
ENFILE	Too many open files in system	系统中打开的文件太多
ENODEV	No such device	所需要的设备不存在
ENOENT	No such file or directory	输入的文件或路径不存在
ENOEXEC	Exec format error	执行的格式错误
ENOLCK	No lock	未锁
ENOLINK	Virtual circuit is gone	实际电路已不存在
ENOMEM	Not enough space	无足够的存储空间
ENOMSG	No message of desired type	并非所需类型的信息
EDOM	Math argument	计算函数参数的域错误
EFAULT	Bad address	错误的地址
EFBIG	File too large	文件太长
EHOSTDOWN	Host is down	主机故障
EHOSTUNREACH	Host is unreachable	主机功能达不到
EIDRM	Identifier removed	标识符丢失
EINPROGRESS	Connection already in progress	连接已在处理中
EINTR	Interrupted system call	中断系统调用
EINVAL	Invalid argument	非法参数

续表 F6

errnum	错误信息串	含义解释
EIO	I/O error	输入/输出错误
EISCONN	Socket is already connected	插口已被连接
EISDIR	Is a directory	路径错误
ELIBACC	Cannot access a needed shared library	要求共享的库不能被访问
ELIBBAD	Accessing a corrupted shared library	要访问的共享库已被破坏
ELIBEXEC	Cannot exec a shared library directly	不可直接执行一个共享库
ELIBMAX	Attempting to link in more shared libraries than system limit	链接的共享库已超出限度
ELIBSCN	<<.lib>> section in a.out corrupted	在*.out 文件中的<.lib>损坏
EMFILE	Too many open files	要打开的文件太多
EMLINK	Too many links	要链接的模块太多
EMSGSIZE	Message too long	信息太长
EMULTIHOP	Multihop attempted	非法的多重接收
ENAMETOOLONG	File or path name too long	文件或路径名太长
ENETDOWN	Network interface not configured	网络接口未配置
ENETUNREACH	Network is unreachable	网络功能达不到
ENFILE	Too many open files in system	系统中打开的文件太多
ENODEV	No such device	所需要的设备不存在
ENOENT	No such file or directory	输入的文件或路径不存在
ENOEXEC	Exec format error	执行的格式错误
ENOLCK	No lock	未锁
ENOLINK	Virtual circuit is gone	实际电路已不存在
ENOMEM	Not enough space	无足够的存储空间
ENOMSG	No message of desired type	并非所需类型的信息
ENONET	Machine is not on the network	所要搜索的机器未上网
ENOPKG	No package	未经压缩
ENOPROTOOPT	Protocol not available	协议不可用
ENOSPC	No space left on device	设备未留有足够的空间
ENOSR	No stream resources	非流资源
ENOSTR	Not a stream	不是一个流
ENOSYS	Function not implemented	函数未被执行
ENOTBLK	Block device required	未接通所需的模块设备
ENOTCONN	Socket is not connected	插口未连接
ENOTDIR	Not a directory	并非路径
ENOTEMPTY	Directory not empty	路径名仍在占用
ENOTSOCK	Socket operation on non-socket	插口连接未成
ENOTSUP	Not supported	不支持
ENOTTY	Not a character device	非字符设备
ENXIO	No such device or address	无此设备或地址
EPERM	Not owner	并非物主
EPIPE	Broken pipe	破坏的流通管道

续表 F6

errnum	错误信息串	含义解释
EPROTO	Protocol error	协议错误
EPROTOTYPE	Protocol wrong type for socket	协议接口警告
EPROTONOSUPPORT	Unknown protocol	未知协议
ERANGE	Result too large	运算结果超出范围
EREMOTE	Resource is remote	资源位置太远
EROFS	Read-only file system	只读文件系统不可写入
ESHUTDOWN	Can't send after socket shutdown	接口关闭后不能传输信息
ESOCKTNOSUPPORT	Socket type not supported	不支持此类接口类型
ESPIPE	Illegal seek	非法搜寻
ESRCH	No such process	无此过程
ESRMNT	Srmount error	高级装配错误
ETIME	Stream ioctl timeout	I/O 流控制超时
ETIMEDOUT	Connection timed out	连接时间太长
ETXTBSY	Text file busy	文本文件在处理中
EXDEV	Cross-device link	设备交叉链接