|  |
| --- |
| string.h文件中函数的详细用法  先说说strlen 这个比较常用的。         格式：strlen (字符数组名)          功能：计算字符串s的(unsigned int型)长度,不包括'\0'在内          说明：返回s的长度，不包括结束符NULL  与sizeof()的区别strlen(char\*)函数求的是字符串的实际长度，它求得方法是从开始到遇到第一个'\0',如果你只定  义没有 给它赋初值，这个结果是不定的，它会从aa首地址一直找下去，直到遇到'\0'停止。  char aa[10];strlen(aa; //结果是不定的  char aa[10]={'\0'}; strlen（aa）; //结果为0  char aa[10]="jun"; strlen（aa）; //结果为3  sizeof  （aa）; //结果为10；  而sizeof()返回的是变量声明后所占的内存数，不是实际长度，此外sizeof不是函数，仅仅是一个操作符，strlen是函数。  1、strcpy 函数名: strcpy 功 能: 拷贝一个字符串到另一个 用 法: char \*strcpy(char \*destin, char \*source); 程序例: #include <stdio.h> #include <string.h> int main(void) { char string[10]; char \*str1 = "abcdefghi"; strcpy(string, str1);    printf("%s\n", string); return 0; } 2、strcat 函数名: strcat 功 能: 字符串拼接函数 用 法: char \*strcat(char \*destin, char \*source); 程序例: #include <string.h> #include <stdio.h> int main(void) { char destination[25]; char \*blank = " ", \*c = "C++", \*Borland = "Borland"; strcpy(destination, Borland); strcat(destination, blank); strcat(destination, c); printf("%s\n", destination); return 0; } 3、strchr 函数名: strchr 功 能: 在一个串中查找给定字符的第一个匹配之处\ 用 法: char \*strchr(char \*str, char c); 程序例: #include <string.h> #include <stdio.h> int main(void) { char string[15]; char \*ptr, c = 'r'; strcpy(string, "This is a string"); ptr = strchr(string, c); if (ptr) printf("The character %c is at position: %d\n", c, ptr-string); else printf("The character was not found\n"); return 0; } 4、strcmp 函数名: strcmp 功 能: 串比较 用 法: int strcmp(char \*str1, char \*str2); 看Asic码，str1>str2，返回值 > 0；两串相等，返回0 程序例: #include <string.h> #include <stdio.h> int main(void) { char \*buf1 = "aaa", \*buf2 = "bbb", \*buf3 = "ccc"; int ptr; ptr = strcmp(buf2, buf1); if (ptr > 0) printf("buffer 2 is greater than buffer 1\n"); else printf("buffer 2 is less than buffer 1\n"); ptr = strcmp(buf2, buf3); if (ptr > 0) printf("buffer 2 is greater than buffer 3\n"); else printf("buffer 2 is less than buffer 3\n"); return 0; } 5、strncmpi 函数名: strncmpi 功 能: 将一个串中的一部分与另一个串比较, 不管大小写 用 法: int strncmpi(char \*str1, char \*str2, unsigned maxlen); 程序例: #include <string.h> #include <stdio.h> int main(void) { char \*buf1 = "BBB", \*buf2 = "bbb"; int ptr; ptr = strcmpi(buf2, buf1); if (ptr > 0) printf("buffer 2 is greater than buffer 1\n"); if (ptr < 0) printf("buffer 2 is less than buffer 1\n"); if (ptr == 0) printf("buffer 2 equals buffer 1\n"); return 0; } 6、strcpy 函数名: strcpy 功 能: 串拷贝 用 法: char \*strcpy(char \*str1, char \*str2); 程序例: #include <stdio.h> #include <string.h> int main(void) { char string[10]; char \*str1 = "abcdefghi"; strcpy(string, str1); printf("%s\n", string); return 0; } 7、strcspn 函数名: strcspn 功 能: 在串中查找第一个给定字符集内容的段 用 法: int strcspn(char \*str1, char \*str2); 程序例: #include <stdio.h> #include <string.h> #include <alloc.h> int main(void) { char \*string1 = "1234567890"; char \*string2 = "747DC8"; int length; length = strcspn(string1, string2); printf("Character where strings intersect is at position %d\n", length); return 0; } 8、strdup 函数名: strdup 功 能: 将串拷贝到新建的位置处 用 法: char \*strdup(char \*str); 程序例: #include <stdio.h> #include <string.h> #include <alloc.h> int main(void) { char \*dup\_str, \*string = "abcde"; dup\_str = strdup(string); printf("%s\n", dup\_str); free(dup\_str); return 0; } 9、stricmp 函数名: stricmp 功 能: 以大小写不敏感方式比较两个串 用 法: int stricmp(char \*str1, char \*str2); 程序例: #include <string.h> #include <stdio.h> int main(void) { char \*buf1 = "BBB", \*buf2 = "bbb"; int ptr; ptr = stricmp(buf2, buf1); if (ptr > 0) printf("buffer 2 is greater than buffer 1\n"); if (ptr < 0) printf("buffer 2 is less than buffer 1\n"); if (ptr == 0) printf("buffer 2 equals buffer 1\n"); return 0; } 10、strerror 函数名: strerror 功 能: 返回指向错误信息字符串的指针 用 法: char \*strerror(int errnum); 程序例: #include <stdio.h> #include <errno.h> int main(void) { char \*buffer; buffer = strerror(errno); printf("Error: %s\n", buffer); return 0; } 11、strcmpi 函数名: strcmpi 功 能: 将一个串与另一个比较, 不管大小写 用 法: int strcmpi(char \*str1, char \*str2); 程序例: #include <string.h> #include <stdio.h> int main(void) { char \*buf1 = "BBB", \*buf2 = "bbb"; int ptr; ptr = strcmpi(buf2, buf1); if (ptr > 0) printf("buffer 2 is greater than buffer 1\n"); if (ptr < 0) printf("buffer 2 is less than buffer 1\n"); if (ptr == 0) printf("buffer 2 equals buffer 1\n"); return 0; } 函数名: strncmp 功 能: 串比较 用 法: int strncmp(char \*str1, char \*str2, int maxlen); 程序例: #include <string.h> #include <stdio.h> int main(void) { char \*buf1 = "aaabbb", \*buf2 = "bbbccc", \*buf3 = "ccc"; int ptr; ptr = strncmp(buf2,buf1,3); if (ptr > 0) printf("buffer 2 is greater than buffer 1\n"); else printf("buffer 2 is less than buffer 1\n"); ptr = strncmp(buf2,buf3,3); if (ptr > 0) printf("buffer 2 is greater than buffer 3\n"); else printf("buffer 2 is less than buffer 3\n"); return(0); } 12、strncpy 函数名: strncpy 功 能: 串拷贝 用 法: char \*strncpy(char \*destin, char \*source, int maxlen); 程序例: #include <stdio.h> #include <string.h> int main(void) { char string[10]; char \*str1 = "abcdefghi"; strncpy(string, str1, 3); string[3] = '\0'; printf("%s\n", string); return 0; } 13、strnicmp 函数名: strnicmp 功 能: 不注重大小写地比较两个串 用 法: int strnicmp(char \*str1, char \*str2, unsigned maxlen); 程序例: #include <string.h> #include <stdio.h> int main(void) { char \*buf1 = "BBBccc", \*buf2 = "bbbccc"; int ptr; ptr = strnicmp(buf2, buf1, 3); if (ptr > 0) printf("buffer 2 is greater than buffer 1\n"); if (ptr < 0) printf("buffer 2 is less than buffer 1\n"); if (ptr == 0) printf("buffer 2 equals buffer 1\n"); return 0; } 14、strnset 函数名: strnset 功 能: 将一个字符串前n个字符都设为指定字符 用 法: char \*strnset(char \*str, char ch, unsigned n); 程序例: #include <stdio.h> #include <string.h> int main(void) { char \*string = "abcdefghijklmnopqrstuvwxyz"; char letter = 'x'; printf("string before strnset: %s\n", string); strnset(string, letter, 13); printf("string after strnset: %s\n", string); return 0; } 15、strpbrk 函数名: strpbrk 功 能: 在串中查找给定字符集中的字符 用 法: char \*strpbrk(char \*str1, char \*str2); 程序例: #include <stdio.h> #include <string.h> int main(void) { char \*string1 = "abcdefghijklmnopqrstuvwxyz"; char \*string2 = "onm"; char \*ptr; ptr = strpbrk(string1, string2); if (ptr) printf("strpbrk found first character: %c\n", \*ptr); else printf("strpbrk didn't find character in set\n"); return 0; } 16、strrchr 函数名: strrchr 功 能: 在串中查找指定字符的最后一个出现 用 法: char \*strrchr(char \*str, char c); 程序例: #include <string.h> #include <stdio.h> int main(void) { char string[15]; char \*ptr, c = 'r'; strcpy(string, "This is a string"); ptr = strrchr(string, c); if (ptr) printf("The character %c is at position: %d\n", c, ptr-string); else printf("The character was not found\n"); return 0; } 17、strrev 函数名: strrev 功 能: 串倒转 用 法: char \*strrev(char \*str); 程序例: #include <string.h> #include <stdio.h> int main(void) { char \*forward = "string"; printf("Before strrev(): %s\n", forward); strrev(forward); printf("After strrev(): %s\n", forward); return 0; } 18、strset 函数名: strset 功 能: 将一个串中的所有字符都设为指定字符 用 法: char \*strset(char \*str, char c); 程序例: #include <stdio.h> #include <string.h> int main(void) { char string[10] = "123456789"; char symbol = 'c'; printf("Before strset(): %s\n", string); strset(string, symbol); printf("After strset(): %s\n", string); return 0; } 19、strspn 函数名: strspn 功 能: 在串中查找指定字符集的子集的第一次出现 用 法: int strspn(char \*str1, char \*str2); 程序例: #include <stdio.h> #include <string.h> #include <alloc.h> int main(void) { char \*string1 = "1234567890"; char \*string2 = "123DC8"; int length; length = strspn(string1, string2); printf("Character where strings differ is at position %d\n", length); return 0; } 20、strstr 函数名: strstr 功 能: 在串中查找指定字符串的第一次出现 用 法: char \*strstr(char \*str1, char \*str2); 程序例: #include <stdio.h> #include <string.h> int main(void) { char \*str1 = "Borland International", \*str2 = "nation", \*ptr; ptr = strstr(str1, str2); printf("The substring is: %s\n", ptr); return 0; } 21、strtod 函数名: strtod 功 能: 将字符串转换为double型值 用 法: double strtod(char \*str, char \*\*endptr); 程序例: #include <stdio.h> #include <stdlib.h> int main(void) { char input[80], \*endptr; double value; printf("Enter a floating point number:"); gets(input); value = strtod(input, &endptr); printf("The string is %s the number is %lf\n", input, value); return 0; } 22、strtok 函数名: strtok 功 能: 查找由在第二个串中指定的分界符分隔开的单词 用 法: char \*strtok(char \*str1, char \*str2); 程序例: #include <string.h> #include <stdio.h> int main(void) { char input[16] = "abc,d"; char \*p; /\* strtok places a NULL terminator in front of the token, if found \*/ p = strtok(input, ","); if (p) printf("%s\n", p); /\* A second call to strtok using a NULL as the first parameter returns a pointer to the character following the token \*/ p = strtok(NULL, ","); if (p) printf("%s\n", p); return 0; } 23、strtol 函数名: strtol 功 能: 将串转换为长整数 用 法: long strtol(char \*str, char \*\*endptr, int base); 程序例: #include <stdlib.h> #include <stdio.h> int main(void) { char \*string = "87654321", \*endptr; long lnumber; /\* strtol converts string to long integer \*/ lnumber = strtol(string, &endptr, 10); printf("string = %s long = %ld\n", string, lnumber); return 0; } 24、strupr 函数名: strupr 功 能: 将串中的小写字母转换为大写字母 用 法: char \*strupr(char \*str); 程序例: #include <stdio.h> #include <string.h> int main(void) { char \*string = "abcdefghijklmnopqrstuvwxyz", \*ptr; /\* converts string to upper case characters \*/ ptr = strupr(string); printf("%s\n", ptr); return 0; } 25、swab 函数名: swab 功 能: 交换字节 用 法: void swab (char \*from, char \*to, int nbytes); 程序例: #include <stdlib.h> #include <stdio.h> #include <string.h> char source[15] = "rFna koBlrna d"; char target[15]; int main(void) { swab(source, target, strlen(source)); printf("This is target: %s\n", target); return 0; 原型：extern char \*strstr(char \*haystack, char \*needle); \*所在头文件：#include <string.h> \*功能：从字符串haystack中寻找needle第一次出现的位置（不比较结束符NULL)。 \*说明：返回指向第一次出现needle位置的指针，如果没找到则返回NULL。 |

26、memcpy

函数原型void \*memcpy(void \*dest, const void \*src, int n);   
从源src所指的内存地址的起始位置开始拷贝n个字节到目标dest所指的内存地址的起始位置中   
所需头文件  
#include <string.h>   
返回值  
函数返回一个指向dest的指针。   
说明  
1.source和destin所指内存区域不能重叠，函数返回指向destin的指针。   
2.与strcpy相比，memcpy并不是遇到'\0'就结束，而是一定会拷贝完n个字节。   
3.如果目标数组destin本身已有数据，执行memcpy（）后，将覆盖原有数据（最多覆盖n）。

       如果要追加数据，则每次执行memcpy后，要将目标数组地址增加到你要追加数据的地址。   
//注意，source和destin都不一定是数组，任意的可读写的空间均可

程序例  
example1  
作用：将s中的字符串复制到字符数组d中。   
// memcpy.c   
#include <stdio.h>   
#include <string.h>   
int main()   
{   
char \*s="Golden Global View";   
char d[20];   
clrscr();   
memcpy(d,s,strlen(s));   
d[strlen(s)]='\0'; //因为从d[0]开始复制，总长度为strlen(s)，d[strlen(s)]置为结束符   
printf("%s",d);   
getchar();   
return 0;   
}   
输出结果：Golden Global View   
example2  
作用：将s中第14个字符开始的4个连续字符复制到d中。(从0开始)   
#include <string.h>   
int main()   
{   
char \*s="Golden Global View";   
char d[20];   
memcpy(d,s+14,4); //从第14个字符(V)开始复制，连续复制4个字符(View)   
//memcpy(d,s+14\*sizeof(char),4\*sizeof(char));也可   
d[4]='\0';   
printf("%s",d);   
getchar();   
return 0;   
}   
输出结果：　View   
example3  
作用：复制后覆盖原有部分数据   
#include <stdio.h>   
#include <string.h>   
int main(void)   
{   
char src[] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";   
char dest[] = "abcdefghijlkmnopqrstuvwxyz0123as6";   
printf("destination before memcpy: %s\n", dest);   
memcpy(dest, src, strlen(src));   
printf("destination after memcpy: %s\n", dest);   
return 0;   
}   
输出结果：   
destination before memcpy:abcdefghijlkmnopqrstuvwxyz0123as6   
destination after memcpy: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*as6