**1、strcmp函数**

strcmp函数是比较两个字符串的大小,返回比较的结果。一般形式是：   
i=strcmp(字符串，字符串);

其中，字符串１、字符串２均可为字符串常量或变量；i  是用于存放比较结果的整型变量。比较结果是这样规定的：   
①字符串1小于字符串2,strcmp函数返回一个负值;  
②字符串1等于字符串2,strcmp函数返回零;  
③字符串1大于字符串2,strcmp函数返回一个正值;

strcmp函数实际上是对字符的ASCII码进行比较，实现原理如下：首先比较两个串的第一个字符，若不相等，则停止比较并得出两个ASCII码大小比较的结果；如果相等就接着 比较第二个字符然后第三个字符等等。无论两个字符串是什么样，strcmp函数最多比较到其中一个字符串遇到结束符'/0'为止，就能得出结果。strcmp算法的可以有多种，不过我觉的可以把这么多算法分为两种，一种是利用减法运算判断结果，另一种是利用比较运算（==）得出结果。

注意:字符串是数组类型而非简单类型，不能用关系运算进行大小比较。

例：

if("ABC">"DEF")  /\*错误的字符串比较\*/  
if(strcmp("ABC","DEF")  /\*正确的字符串比较\*/

**2、strcpy函数**   
strcpy函数用于实现两个字符串的拷贝。一般形式是：   
strcpy（字符中1,字符串2)

其中,字符串1必须是字符串变量,而不能是字符串常量。strcpy函数把字符串2的内容完全复制到字符串1中,而不管字符串1中原先存放的是什么。复制后，字符串2保持不变。   
例:注意,由于字符串是数组类型,所以两个字符串复制不通过赋值运算进行。   
t=s;  /\*错误的字符串复制\*/  
strcpy(t,s);  /\*正确的字符串复制\*/

字符串处理strcpy strcat函数的用法：

1)strcat是用来连接两个字符串的，原型是char \*strcat(char \*dest,char \*src)，

作用是把src所指字符串添加到dest结尾处(覆盖dest结尾处的'\0')并添加'\0'

2) strcpy是用来把字符串拷贝到指定的地方的，

原型是char \*strcpy(char \*dest,const char \*src)，

作用是把从src地址开始且含有NULL结束符的字符串复制到以dest开始的地址空间

注意strcat是从dest的结尾处开始操作的，而strcpy是直接覆盖dest指向的内容。

标准输出流

它展开到一个 FILE\* （“指向 FILE 的指针”）类型的表达式（不一定是常量），

这个表达式指向一个与标准输出流（standard output stream）相关连的 FILE 对象

例子：

#include<stdio.h>

void func(FILE\*p)

{

char\*str="ddd";

fputs(str,p);

}

int main()

{

func(stdout);

return 0;

}

原型：int sprintf( char \*buffer, const char \*format, [ argument] … ); sprintf函数打印到字符串中

usleep()函数是把调用该函数的线程挂起一段时间，单位是微秒（百万分之一秒）。

unlink()函数功能即为删除文件，原型：int unlink(const char \*pathname);

参考链接：

<https://www.runoob.com/cprogramming/c-standard-library-string-h.html>