

C 语言字符串处理函数

C 语言提供了丰富的字符串处理函数，例如字符串的输入、输出、合并、修改、比较、转换、复制、搜索等，使用这些现成的函数可大大减轻编程的负担。

用于输入输出的字符串函数，例如 `printf`、`puts`、`scanf`、`gets` 等，使用时应包含头文件 `stdio.h`，使用其它字符串函数则应包含头文件 `string.h`。

字符串长度函数 `strlen`

`strlen` 是 `string length` 的缩写，用来获得字符串的长度。所谓长度，就是包含多少个字符（不包括字符串结束标志 `'\0'`）。语法格式为：

```
strlen(arrayName);
```

`strlen` 将返回字符串的长度，它是一个整数。请看下面的例子：

```
1.     #include <stdio.h>
2.     #include <string.h>
3.     int main() {
4.         char str[]="C language";
5.         int len = strlen(str);
6.         printf("The lenth of the string is %d\n", len);
7.
8.         return 0;
9.     }
```

运行结果：

```
The lenth of the string is 10
```

需要说明的是，`strlen` 会从字符串的第 0 个字符开始计算，直到遇到字符串结束标志 `'\0'`。将上面代码中的 `str` 改为：

```
char str[]="C \0language";
```

那么输出结果就是：

The lenth of the string is 2

字符串连接函数 strcat

strcat 是 string concatenate 的缩写，意思是将两个字符串拼接在一起，语法格式为：

```
strcat(arrayName1, arrayName2);
```

arrayName1、arrayName2 为需要拼接的字符串。

strcat 将把 arrayName2 连接到 arrayName1 后面，并删去 arrayName1 最后的结束标志 '\0'。这意味着，arrayName1 的长度要足够，必须能够同时容纳 arrayName1 和 arrayName2，否则会越界。

strcat 返回值为 arrayName1 的首地址。请看下面的例子：

```
1.      #include <stdio.h>
2.      #include <string.h>
3.      int main() {
4.          char str1[40]="My name is ";
5.          char str2[20];
6.          printf("Input your name:");
7.          gets(str2);
8.          strcat(str1, str2);
9.          puts(str1);
10.
11.         return 0;
12.     }
```

运行结果：

Input your name:xiao p

My name is xiao p

字符串复制函数 strcpy

strcpy 是 string copy 的缩写，意思是字符串复制，语法格式为：

```
strcpy(arrayName1, arrayName2);
```

strcpy 会把 arrayName2 中的字符串拷贝到 arrayName1 中，串结束标志 '\0' 也一同拷贝。请看下面的例子：

```
1.     #include <stdio.h>
2.     #include <string.h>
3.     int main() {
4.         char str1[15], str2[]="C Language";
5.         strcpy(str1, str2);
6.         puts(str1);
7.         printf("\n");
8.
9.         return 0;
10.    }
```

运行结果：

```
C Language
```

strcat 要求 arrayName1 要有足够的长度，否则不能全部装入所拷贝的字符串。

字符串比较函数 strcmp

strcmp 是 string compare 的缩写，意思是字符串比较，语法格式为：

```
strcmp(arrayName1, arrayName2);
```

arrayName1 和 arrayName2 是需要比较的两个字符串。

字符本身没有大小之分，strcmp() 是以各个字符在 ASCII 码表上对应的数值进行比较的。strcmp() 首先将 arrayName1 中第 0 个字符的 ASCII 码值减去 arrayName2 中第 0 个字符的 ASCII 码值，若差值为 0，则说明两个字符相同，再继续比较下个字符，若差值不为 0 则将差值返回。例如字符串"Ac"和"ba"比较则会返回字符"A"(65)和'b'(98)的差值(- 33)。

返回值：若 arrayName1 和 arrayName2 相同，则返回 0；若 arrayName1 大于

arrayName2 则返回大于 0 的值 若 arrayName1 小于 arrayName2 则返回小于 0 的值。

【示例】对 4 组字符串进行比较。

```
1.      #include <string.h>
2.      main() {
3.          char *a = "aBcDeF";
4.          char *b = "AbCdEf";
5.          char *c = "aacdef";
6.          char *d = "aBcDeF";
7.          printf("strcmp(a, b) : %d\n", strcmp(a, b));
8.          printf("strcmp(a, c) : %d\n", strcmp(a, c));
9.          printf("strcmp(a, d) : %d\n", strcmp(a, d));
10.     }
```

运行结果：

strcmp(a, b) : 32

strcmp(a, c) :-31

strcmp(a, d) : 0