

C 标准库 - <string.h>

简介

`string.h` 头文件定义了一个变量类型、一个宏和各种操作字符数组的函数。

库变量

下面是头文件 `string.h` 中定义的变量类型：

序号	变量 & 描述
1	size_t 这是无符号整数类型，它是 sizeof 关键字的结果。

库宏

下面是头文件 `string.h` 中定义的宏：

序号	宏 & 描述
1	NULL 这个宏是一个空指针常量的值。

库函数

下面是头文件 `string.h` 中定义的函数：

序号	函数 & 描述
1	<code>void *memchr(const void *str, int c, size_t n)</code> 在参数 <code>str</code> 所指向的字符串的前 <code>n</code> 个字节中搜索第一次出现字符 <code>c</code> （一个无符号字符）的位置。
2	<code>int memcmp(const void *str1, const void *str2, size_t n)</code> 把 <code>str1</code> 和 <code>str2</code> 的前 <code>n</code> 个字节进行比较。
3	<code>void *memcpy(void *dest, const void *src, size_t n)</code> 从 <code>src</code> 复制 <code>n</code> 个字符到 <code>dest</code> 。
4	<code>void *memmove(void *dest, const void *src, size_t n)</code> 另一个用于从 <code>str2</code> 复制 <code>n</code> 个字符到 <code>str1</code> 的函数。
5	<code>void *memset(void *str, int c, size_t n)</code> 复制字符 <code>c</code> （一个无符号字符）到参数 <code>str</code> 所指向的字符串的前 <code>n</code> 个字符。
6	<code>char *strcat(char *dest, const char *src)</code>

	把 <i>src</i> 所指向的字符串追加到 <i>dest</i> 所指向的字符串的结尾。
7	<u>char *strncat(char *dest, const char *src, size_t n)</u> 把 <i>src</i> 所指向的字符串追加到 <i>dest</i> 所指向的字符串的结尾，直到 <i>n</i> 字符长度为止。
8	<u>char *strchr(const char *str, int c)</u> 在参数 <i>str</i> 所指向的字符串中搜索第一次出现字符 <i>c</i> （一个无符号字符）的位置。
9	<u>int strcmp(const char *str1, const char *str2)</u> 把 <i>str1</i> 所指向的字符串和 <i>str2</i> 所指向的字符串进行比较。
10	<u>int strncmp(const char *str1, const char *str2, size_t n)</u> 把 <i>str1</i> 和 <i>str2</i> 进行比较，最多比较前 <i>n</i> 个字节。
11	<u>int strcoll(const char *str1, const char *str2)</u> 把 <i>str1</i> 和 <i>str2</i> 进行比较，结果取决于 LC_COLLATE 的位置设置。
12	<u>char *strcpy(char *dest, const char *src)</u> 把 <i>src</i> 所指向的字符串复制到 <i>dest</i> 。
13	<u>char *strncpy(char *dest, const char *src, size_t n)</u> 把 <i>src</i> 所指向的字符串复制到 <i>dest</i> ，最多复制 <i>n</i> 个字符。
14	<u>size_t strcspn(const char *str1, const char *str2)</u> 检索字符串 <i>str1</i> 开头连续有几个字符都不含字符串 <i>str2</i> 中的字符。
15	<u>char *strerror(int errnum)</u> 从内部数组中搜索错误号 <i>errnum</i> ，并返回一个指向错误消息字符串的指针。
16	<u>size_t strlen(const char *str)</u> 计算字符串 <i>str</i> 的长度，直到空结束字符，但不包括空结束字符。
17	<u>char *strpbrk(const char *str1, const char *str2)</u> 检索字符串 <i>str1</i> 中第一个匹配字符串 <i>str2</i> 中字符的字符，不包含空结束字符。也就是说，依次检验字符串 <i>str1</i> 中的字符，当被检验字符在字符串 <i>str2</i> 中也包含时，则停止检验，并返回该字符位置。
18	<u>char *strrchr(const char *str, int c)</u> 在参数 <i>str</i> 所指向的字符串中搜索最后一次出现字符 <i>c</i> （一个无符号字符）的位置。
19	<u>size_t strspn(const char *str1, const char *str2)</u> 检索字符串 <i>str1</i> 中第一个不在字符串 <i>str2</i> 中出现的字符下标。
20	<u>char *strstr(const char *haystack, const char *needle)</u>

	在字符串 <i>haystack</i> 中查找第一次出现字符串 <i>needle</i> (不包含空结束字符) 的位置。
21	<code>char *strtok(char *str, const char *delim)</code> 分解字符串 <i>str</i> 为一组字符串 , <i>delim</i> 为分隔符。
22	<code>size_t strxfrm(char *dest, const char *src, size_t n)</code> 根据程序当前的区域选项中的 LC_COLLATE 来转换字符串 src 的前 n 个字符 , 并把它们放置在字符串 dest 中。

✎ 点我分享笔记