

C 库函数 - strcpy()

 [C 标准库 - <string.h>](#)

描述

C 库函数 `char *strcpy(char *dest, const char *src)` 把 `src` 所指向的字符串复制到 `dest`。
需要注意的是如果目标数组 `dest` 不够大，而源字符串的长度又太长，可能会造成缓冲溢出的情况。

声明

下面是 `strcpy()` 函数的声明。

```
char *strcpy(char *dest, const char *src)
```

参数

- `dest` -- 指向用于存储复制内容的目标数组。
- `src` -- 要复制的字符串。

返回值

该函数返回一个指向最终的目标字符串 `dest` 的指针。

实例

下面的实例演示了 `strcpy()` 函数的用法。

实例 1

```
#include <stdio.h>
#include <string.h>

int main()
{
    char src[40];
    char dest[100];

    memset(dest, '\0', sizeof(dest));
    strcpy(src, "This is runoob.net");
    strcpy(dest, src);
    printf("最终的目标字符串: %s\n", dest);

    return(0);
}
```

让我们编译并运行上面的程序，这将产生以下结果：

```
最终的目标字符串:  This is runoob.net
```

实例 2

```
#include <stdio.h>
#include <string.h>

int main ()
{
    char str1[]="Sample string";
    char str2[40];
    char str3[40];
    strcpy (str2,str1);
    strcpy (str3,"copy successful");
    printf ("str1: %s\nstr2: %s\nstr3: %s\n",str1,str2,str3);
    return 0;
}
```

让我们编译并运行上面的程序，这将产生以下结果：

```
str1: Sample string
str2: Sample string
str3: copy successful
```

 [C 标准库 - <string.h>](#)

实例1中，感觉把sizeof(dest)写成sizeof(dest)/sizeof(char)更严谨些。

实例中的dest数组存放char类型元素，sizeof(dest)的值为100，恰好为数组的大小。

但是如果dest是int类型，一般而言sizeof(dest)的值为400，而不是数组的大小。

回楼上，可是memset就是逐字节的啊，不是数组元素个数

strcpy(char* dest,char* src); 字符串拷贝

memset(void * dest ,int ch, int size_t); 初始化赋值

memcpy(void * dest, void * src, unsigned n); 内存拷贝
