

C 库函数 - memcpy()

 [C 标准库 - <string.h>](#)

描述

C 库函数 `void *memcpy(void *str1, const void *str2, size_t n)` 从存储区 `str2` 复制 `n` 个字节到存储区 `str1`。

声明

下面是 `memcpy()` 函数的声明。

```
void *memcpy(void *str1, const void *str2, size_t n)
```

参数

- `str1` -- 指向用于存储复制内容的目标数组，类型强制转换为 `void*` 指针。
- `str2` -- 指向要复制的数据源，类型强制转换为 `void*` 指针。
- `n` -- 要被复制的字节数。

返回值

该函数返回一个指向目标存储区 `str1` 的指针。

实例

下面的实例演示了 `memcpy()` 函数的用法。

实例

```
// 将字符串复制到数组 dest 中
#include <stdio.h>
#include <string.h>

int main ()
{
    const char src[50] = "http://www.runoob.net";
    char dest[50];

    memcpy(dest, src, strlen(src)+1);
    printf("dest = %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：

```
dest = http://www.runoob.net
```

将 `s` 中第 11 个字符开始的 6 个连续字符复制到 `d` 中:

实例

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

int main()
{
    char *s="http://www.runoob.net";
    char d[20];
    memcpy(d, s+11, 6);// 从第 11 个字符(r)开始复制，连续复制 6 个字符(runoob)
    // 或者 memcpy(d, s+11*sizeof(char), 6*sizeof(char));
    d[6]='\0';
    printf("%s", d);
    return 0;
}
```

让我们编译并运行上面的程序，这将产生以下结果：

```
runoob
```

覆盖原有部分数据:

实例

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

int main(void)
{
    char src[] = "***";
    char dest[] = "abcdefg";
    printf("使用 memcpy 前: %s\n", dest);
    memcpy(dest, src, strlen(src));
    printf("使用 memcpy 后: %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：

```
使用 memcpy 前: abcdefg
使用 memcpy 后: ***defg
```