

规格严格 功夫到家



计算思维与实践

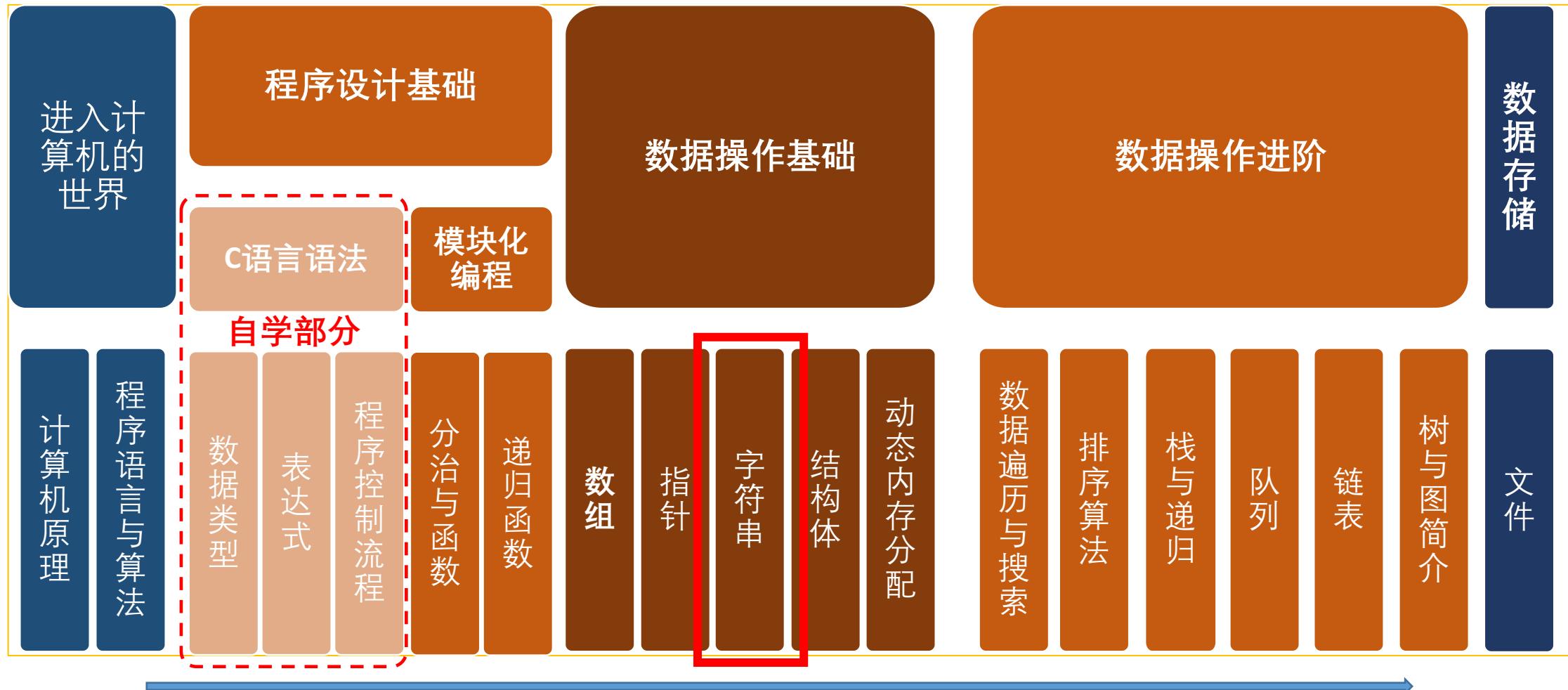
哈尔滨工业大学（深圳）
计算机科学与技术学院
大数据技术中心
张保权

课件.版权：哈尔滨工业大学.苏小红 sxh@hit.edu.cn

版权所有，违者必究

课程内容安排

2



程序设计思想与数据操作方法融会贯通，内容由浅入深

第九讲 学习内容

3

9. 1 字符串常量

9. 2 字符数组

9. 3 字符指针

9. 4 字符串处理函数

9. 5 向函数传递字符串

9. 6 从函数返回字符串指针



第九讲 学习内容

4

9. 1 字符串常量

9. 2 字符数组

9. 3 字符指针

9. 4 字符串处理函数

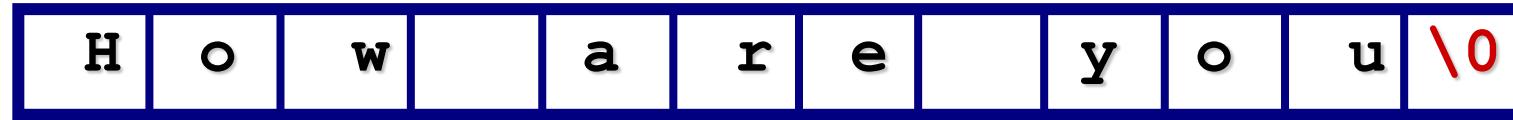
9. 5 向函数传递字符串

9. 6 从函数返回字符串指针



9.1 字符串常量

5



一串以'\0'结尾的字符串在C语言中被看作字符串

"How are you"

用双引号括起的一串字符是字符串常量，C语言自动为其添加'\0'结束符

第九讲 学习内容

6

9. 1 字符串常量

9. 2 字符数组

9. 3 字符指针

9. 4 字符串处理函数

9. 5 向函数传递字符串

9. 6 从函数返回字符串指针



9.2 字符数组

7

C语言没有提供专门的字符串数据类型

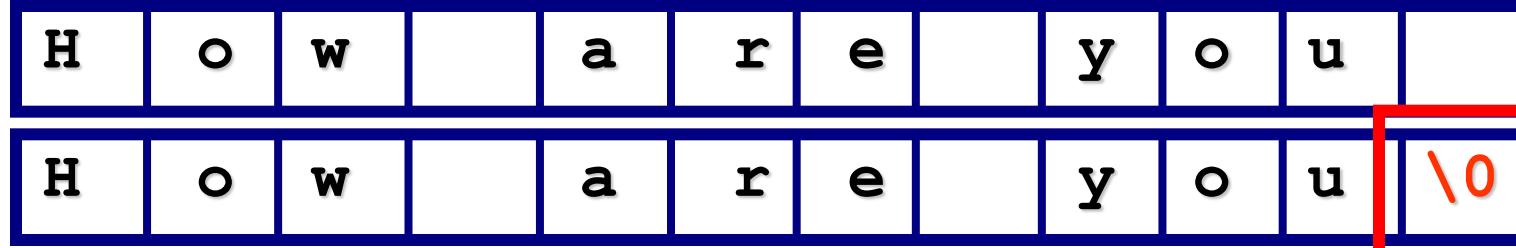
使用字符数组和字符指针来处理字符串

- **字符数组**

是字符数组，但不一定代表字符串

- 每个元素都是字符类型的数组

```
char str[80];
```



数组的最后一个元素必须是'\0'才表示字符串

9.2 字符数组

8

■ 字符数组的初始化

- 用字符常量的初始化列表对数组初始化
 - `char str[6] = {'C', 'h', 'i', 'n', 'a', '\0'};`
- 用字符串常量直接对数组初始化
 - `char str[6] = {"China"};`
 - `char str[6] = "China";`
 - `char str[] = "China";`

第九讲 学习内容

9

9. 1 字符串常量

9. 2 字符数组

9. 3 字符指针

9. 4 字符串处理函数

9. 5 向函数传递字符串

9. 6 从函数返回字符串指针



9.3 字符指针

10

还可以用字符指针来指向一个字符串
如果让字符指针指向一个**字符串常量**

```
char *pStr = "Hello China";
```



字符指针就是指向字符串首地址的指针

9.3 字符指针

11



9.3 字符指针

12



```
char str[] = "Hello China";
```

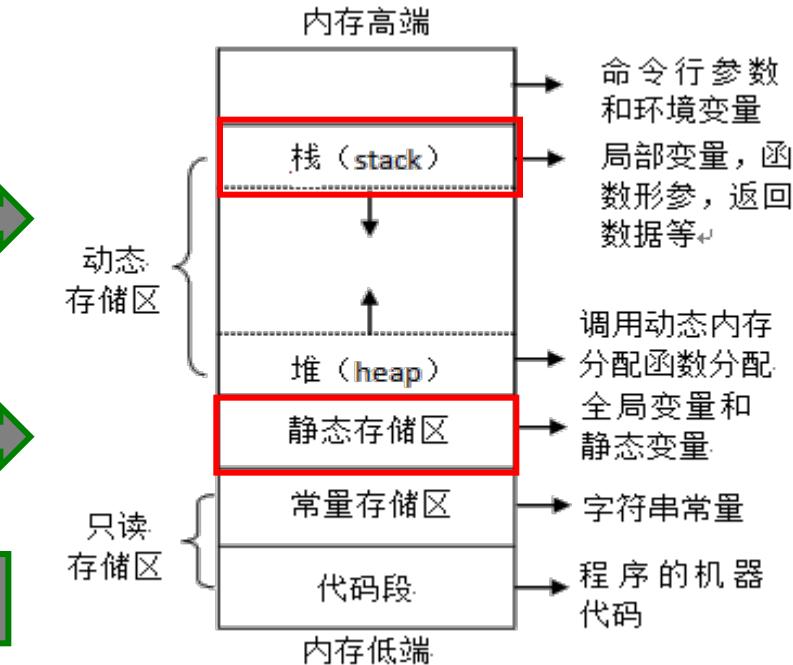
~~str = "Hello China";~~

函数内定义，保存在动态存储区

函数外定义，或定义为静态数组，
字符串保存在静态存储区

数组名 str 的值不可修改，是地址常量

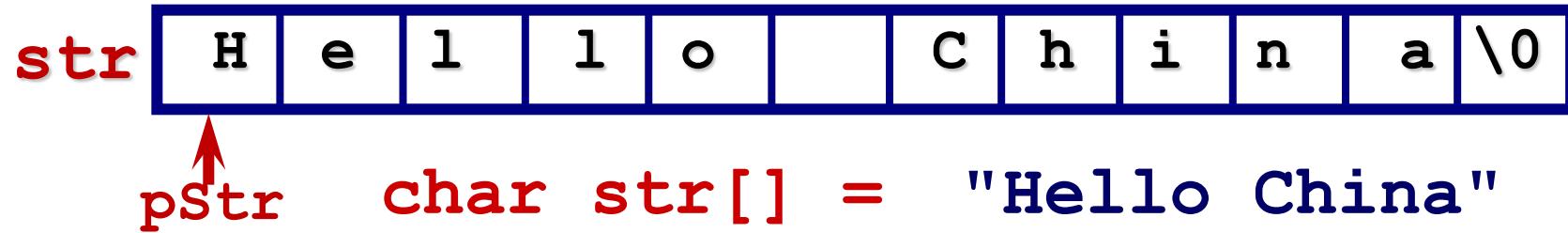
数组中存储的字符可以被修改



str[0] = 'W';

9.3 字符指针

13

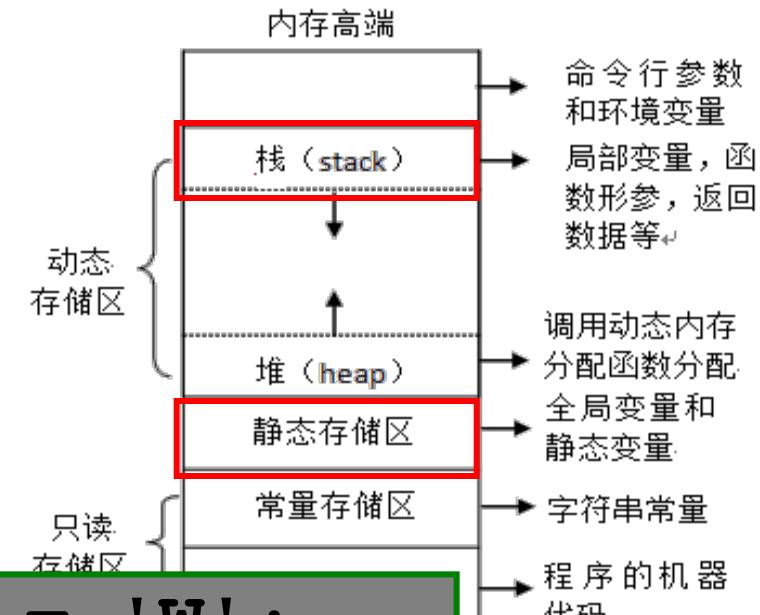


```
char *pStr;
pStr = str;
pStr = &str[0];
```

pStr的值（指向）可以被修改，它所指向的字符串也可以被修改

```
*pStr = 'W';
```

→ **str[0] = 'W';**
 → **pStr[0] = 'W';**



9.3 字符指针

14

正确使用和区分字符数组和字符指针

须牢记以下基本原则：

- 明确字符串被保存到了哪里
- 明确字符指针指向了哪里

9.3 字符指针

15

第九讲 学习内容

16

9. 1 字符串常量

9. 2 字符数组

9. 3 字符串访问与输入输出

9. 4 字符串处理函数

9. 5 向函数传递字符串

9. 6 从函数返回字符串指针



9.4 字符串处理函数

17

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

strlen(字符串) ;

string length

strcpy(目的字符串, 源字符串) ;

string copy

strcat(目的字符串, 源字符串) ;

string combination

strcmp(字符串1, 字符串2) ;

string comparison

9.4.1 字符串处理函数—计算字符串长度

18

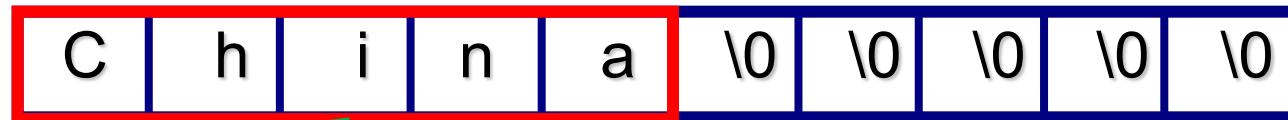
```
#include <string.h>
```

strlen(字符串);

string length

```
char str[10] = {"China"};  
printf("%d", strlen(str));
```

打印结果是 5, 6, 还是10?



不包括\0的实际字符的个数



9.4.1 字符串处理函数—计算字符串长度

19

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

strlen(字符串);

string length

```
len = strlen(str);
for (i=0; i<len; i++)
{
    putchar(str[i]);
}
putchar('\n');
```

用长度控制字符串输出

9.4.2 字符串处理函数-字符串复制

20

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

strcpy(目的字符串, 源字符串);

string copy

字符串能否用=整体复制?

str2 = str1;



strcpy(str2, str1);

注意复制的方向!
str2必须足够大!



9.4.3字符串处理函数-字符串连接

21

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

strcat(目的字符串, 源字符串);

string
combination

strcat(str2, str1);

str2必须足够大!

H	e			o	\0	\0	\0	\0	\0	\0	\0
---	---	--	--	---	----	----	----	----	----	----	----

C	h	i	n	a	\0
---	---	---	---	---	----

H	e			o	C	h	i	n	a	\0	\0
---	---	--	--	---	---	---	---	---	---	----	----

9.4.4字符串处理函数-字符串比较

22

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

strcmp(字符串1, 字符串2);

string
comparison

字符串能否用>,<,==比较大小?

if (str2 == str1)



if (strcmp(str2, str1) == 0)



9.4.4字符串处理函数-字符串比较

23

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

strcmp(字符串1,字符串2);

string
comparison

字符串是如何比较大小的?

compare
computer

判断compare 小于 computer?

strcmp(str1, str2) < 0为真

当出现第一对不相等的字符时，就由这两个字符决定所在字符串的大小，返回其ASCII码比较的结果值



【例4】对国名在字典中的顺序对其进行排序

24

```
#include <stdio.h>
#include <string.h>
#define MAX_LEN 10 /* 字符串最大长度 */
#define N 150 /* 字符串个数 */
void SortString(char str[][MAX_LEN], int n);
int main()
{
    int i, n;
    char name[N][MAX_LEN]; /* 定义二维字符数组 */
    printf("How many countries?");
    scanf("%d", &n);
    getchar(); /* 读走输入缓冲区中的回车符 */
    printf("Input their names:\n");
    for (i=0; i<n; i++)
    {
        gets(name[i]); /* 输入n个字符串 */
    }
    SortString(name, n); /* 字符串按字典顺序排序 */
    printf("Sorted results:\n");
    for (i=0; i<n; i++)
    {
        puts(name[i]); /* 输出排序后的n个字符串 */
    }
    return 0;
}
```

```
How many countries? 5 ✓
Input their names:
America ✓
England ✓
Australia ✓
Sweden ✓
Finland ✓
Sorted results:
America
Australia
England
Finland
Sweden
```

【例4】对国名在字典中的顺序对其排序

25

```
void SortString(char str[][MAX_LEN], int n)
{
    int i, j;
    char temp[MAX_LEN];
    for (i=0; i<n-1; i++)
    {
        for (j=i+1; j<n; j++)
        {
            if (strcmp(str[j], str[i]) < 0)
            {
                strcpy(temp, str[i]);
                strcpy(str[i], str[j]);
                strcpy(str[j], temp);
            }
        }
    }
}
```

9.4.7 字符串处理函数-其他

26

函数功能	函数调用的一般形式	功能描述及其说明
求字符串长度	<code>strlen(str);</code>	由函数值返回字符串 str 的实际长度, 即不包括'\0'在内的实际字符的长度
字符串拷贝	<code>strcpy(str1,str2);</code>	将字符串 str2 复制到字符数组 str1 中, 这里应确保字符数组 str1 的大小足以存放得下字符串 2
字符串比较	<code>strcmp(str1,str2);</code>	<p>比较字符串 str1 和字符串 str2 的大小, 结果分为 3 种情况:</p> <ul style="list-style-type: none"> 当 str1 大于 str2 时, 函数返回值大于 0 当 str1 等于 str2 时, 函数返回值等于 0 当 str1 小于 str2 时, 函数返回值小于 0 <p>字符串的比较方法为: 对两个字符串从左至右按字符的 ASCII 码值大小逐个字符相比较, 直到出现不同的字符或遇到'\0'为止</p>
字符串连接	<code>strcat(str1,str2);</code>	将字符串 str2 添加到字符数组 str1 中的字符串的末尾, 字符数组 str1 中的字符串结束符被字符串 str2 的第一个字符覆盖, 连接后的字符串存放在字符数组 str1 中, 函数调用后返回字符数组 str1 的首地址。这里, 字符数组 str1 应定义得足够大, 以便能存放连接后的字符串
“n 族”字符串拷贝	<code>strncpy(str1,str2,n)</code>	将字符串 str2 的至多前 n 个字符拷贝到字符数组 str1 中
“n 族”字符串比较	<code>strncmp(str1,str2,n)</code>	函数 strncmp(str1,str2,n) 的功能与函数 strcmp(str1,str2) 类似, 它们的不同之处在于, 前者最多比较 n 个字符
“n 族”字符串连接	<code>strncat(str1,str2,n)</code>	将字符串 str2 的至多前 n 个字符添加到字符串 str1 的末尾。str1 的字符串结束符被 str2 中的第一个字符覆盖

第九讲 学习内容

27

9. 1 字符串常量

9. 2 字符数组

9. 3 字符串访问与输入输出

9. 4 字符串处理函数

9. 5 向函数传递字符串

9. 6 从函数返回字符串指针



9.5 向函数传递字符串

28

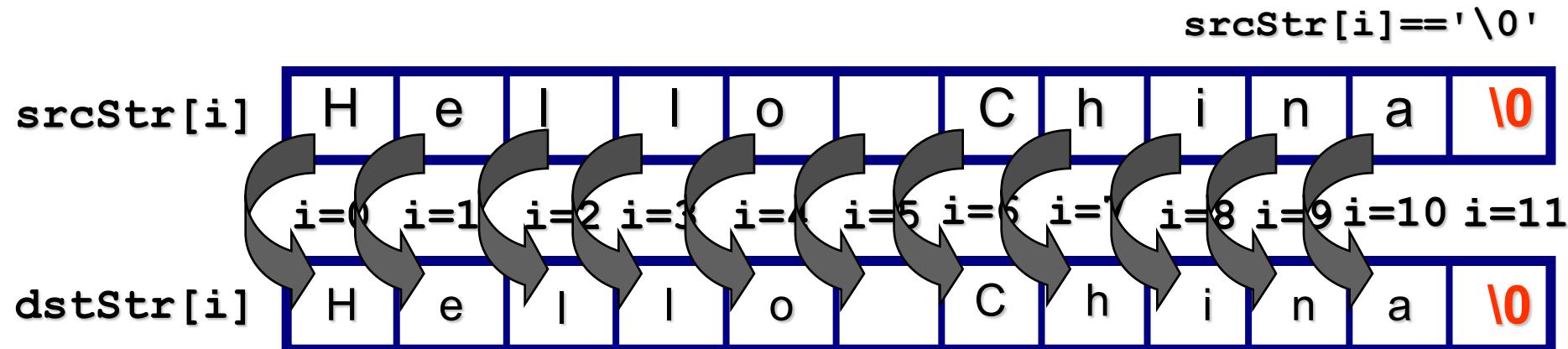
- 向函数传递字符串时
 - 既可用**字符数组**作函数参数
 - 也可用**字符指针**作函数参数
- 传地址调用
 - 传字符串的首地址，而非字符串中的全部字符



【例5】字符串拷贝—用字符数组编程

29

```
void MyStrcpy(char dstStr[], char srcStr[])
{
    int i = 0;
    while (srcStr[i] != '\0')
    {
        dstStr[i] = srcStr[i];
        i++;
    }
    dstStr[i] = '\0';
}
```

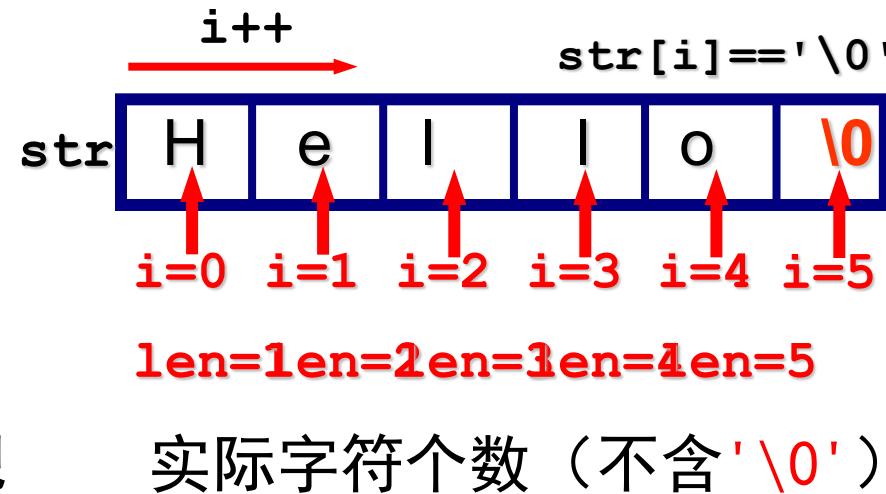


【例6】计算实际字符个数

30

```
unsigned int MyStrlen(char str[])
{
    int i;
    unsigned int len = 0;
    for (i=0; str[i]!='\0'; i++)
    {
        len++;
    }
    return len;
}
```

用字符数组实现



本讲小结

31

- 明确字符串被保存到了哪里
- 向函数传递字符串
 - 向函数传递字符数组
- 字符串处理函数

