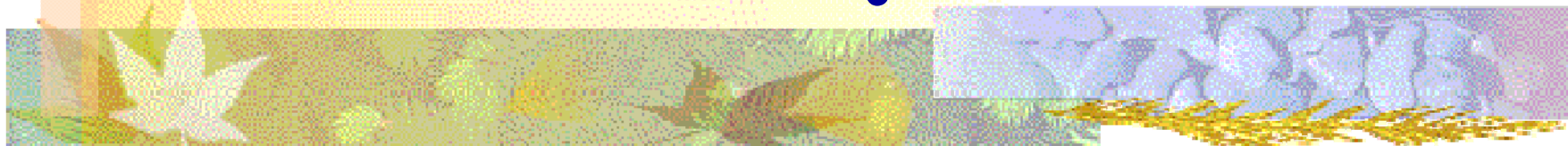


# 程序设计语言C

## 指针与字符、字符串



河海大学

计算机与信息学院

王继民

[wangjimin@hhu.edu.cn](mailto:wangjimin@hhu.edu.cn)

# 字符数组

- 字符数组是以字符为单位存储，处理时，也是按照字符单独处理。字符数组的元素类型为字符的数组。
- `char ch[10];`
- `ch[0] = 'a'; ch[2] = 'v';`
- 每个元素就是一个字符变量。
- `char ch[10]= {'c', 'h', 'i'} ;`

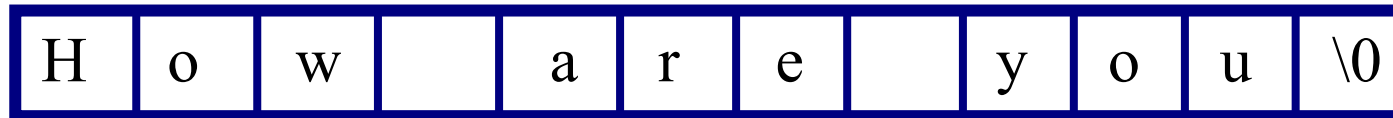


# 字符串

- 在信息系统中，一般如地址，姓名等都是将很多字符作为一个有意义的整体来处理的。
- C语言中，字符串是在内存中连续存储的，并且以'\0'字符结束。
- C中的字符串处理，大部分函数都是以特殊的'\0'作为字符串的结束标志。

# 字符串常量

- 书写字符串常量时，以双引号括起来，如：
- **"How are you"**
- **"南京市江宁区佛城西路8号"**



一串以' \0' 结尾的字符在C语言中被看作字符串

# 字符串变量

## C语言中采用字符数组存储字符串变量

- 用字符型数据对字符数组进行初始化

```
char str[] = {'C','h','i','n','a','\0'};
```

```
printf("%s",str);
```

```
char str1[] = {'C','h','i','n','a'}; //区别?
```

```
printf("%s",str1); //字符数组不一定是字符串
```

- 用字符串常量直接对字符数组初始化

```
— char str[6] = {"China"};
```

```
— char str[6] = "China";
```

2019-11-20 `sizeof(str); // ??`

# 字符数组的输入输出

有&操作，因为  
数组名就是地址

```
char str[10];
```

一次性输入输出

```
scanf("%s", str);  
printf("%s", str);
```

空格、回车或跳格  
(Tab) 符作为输入  
数据的分隔符，因  
而不能被读入，输  
入遇到这些字符时，  
系统认为字符串输  
入结束；系统自动  
加入\0结束

```
gets(str);  
puts(str);
```

可以输入带空格  
的字符串



# 指针与字符数组、字符串

区分以下两种定义有何区别？

```
char str[]="I love CHINA!";  
char *str2 = "I love CHINA!";
```

**str:**系统根据初始化字符串的大小给字符数组str分配空间，同时str代表了该地址，是一个常量。

**str2:**系统分配空间存储初始化字符串，并将该存储空间的起始地址返回存放到str2中，str2是一个变量。

# 指针与字符数组、字符串

定义一个字符指针变量并初始化，然后输出它指向的字符串。

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

```
int main( )
```

```
{    char *str="I love CHINA!";  
    printf("%s\n", str); //I love CHINA! 整体输出  
    printf("%s, %s", str[0],str[2] ); //I  数组方式  
    printf("%s,%s ", *str, *(str+1)); // 指针方式  
    return 0;  
}
```

对字符串中字符的存取，可以用下标方法，也可以用指针方法。





## 分析以下各行代码的内存分配操作

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

```
int main(){
```

```
    char str[]="I love CHINA!";
```

```
    char *str2 = "I love CHINA!";
```

```
    //...
```

```
    str = "Hello world"; //错误,因为str是常量
```

```
    str2 = "Hello world"; //正确, 系统再次分配空间存  
储"Hello world",并将该地址的起始地址返回给str2
```

```
}
```

# 字符串处理函数

■ `#include <string.h>`

`strlen(字符串);`

`strcpy(目的字符串,源字符串);`

`strcat(目的字符串,源字符串);`

`strcmp(字符串1,字符串2);`

# 计算字符串长度

■ #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?
```



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

不包括\0的实际字符的个数

# 用字符数组实现的计算字符串长度函数

■ #include <string.h>

strlen(字符串);

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

# 字符串拷贝

■ `#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!

# 字符串拷贝

■ `#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;`



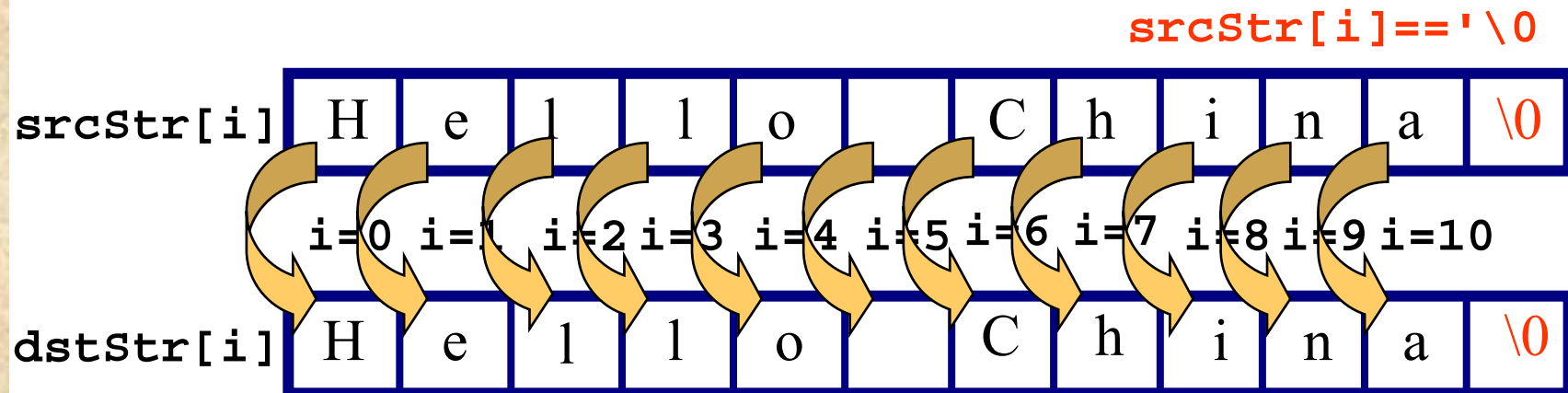
`strcpy(str2, str1);`

想想这个函数是如何实现字符串拷贝的?



# 用字符数组编程实现的字符串拷贝函数

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



# 字符串连接

■ `#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	l	l	o	\0	\0	\0	\0	\0	\0	\0
---	---	---	---	---	----	----	----	----	----	----	----

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

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

# 字符串比较

■ `#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)
```

# 字符串比较

■ `#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

`compare < computer` 表示为  
`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

```
#include <stdio.h>
#include <string.h>
#define ARR_SIZE 80

main()
{
    int    n, num;
    char   str[ARR_SIZE], min[ARR_SIZE];

    printf("Please enter five names:\n");
    gets(str);
    strcpy(min, str);
    for (n=1; n<5; n++)
    {
        gets(str);
        if (strcmp(str, min) < 0)
        {
            strcpy(min, str);
        }
    }
    printf("The min is:");
    puts(min);
}
```

- 从键盘任意输入5个学生的姓名，编程找出并输出按字典顺序排在最前面的学生姓名
- 等价于求最小字符串



# 作 业

- 1.熟悉C语言中的字符串处理函数。
- 2.继续完成员工信息管理系统。