



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

文字信息的表示与处理

- ◆ 3、字符串的处理
- ◆ 4、字符串与数值之间的转换

6.4.4 字符串数组

- ▶ 可以利用二维字符数组来定义字符串数组，定义形式为：

```
char 字符串数组名[常量表达式1][常量表达式2], .....;
```

- ▶ 示例

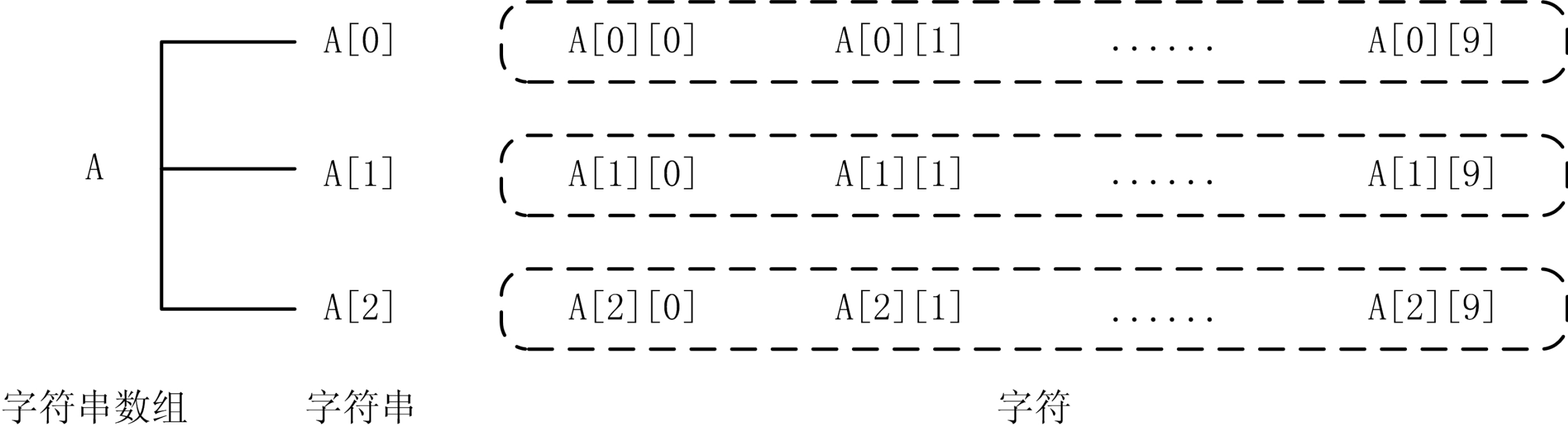
```
char A[3][10]; //定义二维字符数组
```

6.4.4 字符串数组

- ▶ 所谓字符串数组是指这样的一个集合，每个元素都是一个字符串。如上面的定义有三个元素，分别是A[0]、A[1]、A[2]，每个元素都是一个字符串。由于字符串是一维数组，那么字符串数组就应该是二维数组。

6.4.4 字符串数组

图6.4 字符串数组内存结构



6.4.4 字符串数组

- ▶ 显然，字符集合组成字符串，字符串集合组成字符串数组。如果用字符串表示一句话、一行文字的话，那么字符串数组就可以表示多行文字、一段文字或一篇文章。

6.4.4 字符串数组

- ▶ 字符串数组的初始化可以采用二维数组初始化的形式，但采用字符串常量形式会更简洁，例如：

```
char A[3][20]={{"C++"}, {"JAVA"}, {"BASIC"}};  
//字符串数组二维初始化形式
```

6.4.4 字符串数组

- ▶ 除外面的大括号不能省略外，里面的均可省略，例如：

```
char A[3][20]={"C++", "JAVA", "BASIC"};  
//字符串数组一维初始化形式
```


6.4.4 字符串数组

- ▶ 按照多维数组初始化要求，第1维可由编译器自动确定，其余必须给定数组长度，如：

```
char A[3][20]={"C++", "JAVA", "BASIC"};  
//字符串数组一维初始化形式
```

- ▶ 一般情况下，都是按估计的最大字符串长度给定的。

6.4.4 字符串数组

- ▶ 字符串数组的输入和输出按字符串方式来进行，例如

```
char A[3][80]; //定义字符串数组，有三个字符串
scanf("%s",A[0]); //输入第0个字符串
gets(A[1]); //输入第1个字符串
printf("%s",A[0]); //输出第0个字符串
puts(A[1]); //输出第1个字符串
```

6.4.5 字符串处理函数

- ▶ C语言标准库提供了很多有用的字符串处理函数，其头文件为string.h。

6.4.5 字符串处理函数

- ▶ (1) 字符串复制函数strcpy (string copy)

```
char *strcpy(char *s1, const char *s2);
```

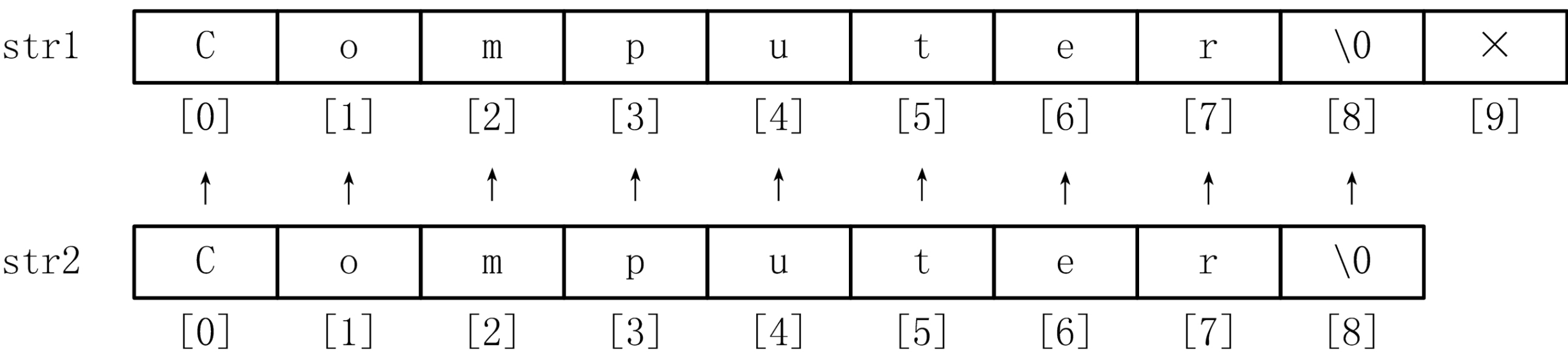
- ▶ strcpy函数将s2中的字符串复制到s1中，包括空字符。s1是字符数组或指向字符数组的指针，其长度应该足够大，以便能容纳被复制的字符串；s2可以是字符串常量、字符数组或指向字符数组的指针。

6.4.5 字符串处理函数

► 例如

```
char str1[10],str2[]="Computer";  
strcpy(str1,str2); //复制str2到str1
```

► 运行过程如图所示。



6.4.5 字符串处理函数

- ▶ 可以复制字符串常量。例如

```
strcpy(str1, "Java");
```

6.4.5 字符串处理函数

- ▶ (2) 字符串复制函数strncpy

```
char *strncpy(char *s1, const char *s2, size_t n);
```

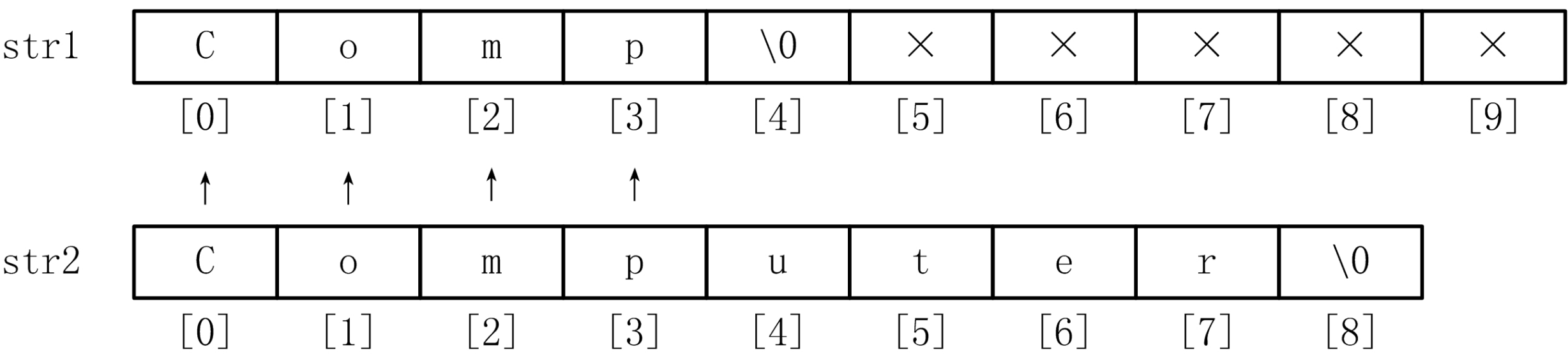
- ▶ strncpy将s2中不超过n个字符的字符串复制到s1中，其他与strcpy函数类似。

6.4.5 字符串处理函数

► 例如

```
char str1[10], str2[]="Computer";  
strncpy(str1,str2,4); //复制str2到str1, 最多4个字符
```

► 运行过程如图所示。



6.4.5 字符串处理函数

- ▶ 如果s2字符串长度未达到n个，则复制整个s2，其余用空字符填充直到n个。例如

```
strncpy(str1, "Java", 8);
```

- ▶ 执行后str1的存储形式为

str1

J	a	v	a	\0	\0	\0	\0	×	×
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

6.4.5 字符串处理函数

- ▶ strncpy复制后可能会使s1没有空字符结束。例如

```
strncpy(str1, "Programming Language", 10);
```

- ▶ 执行后str1的存储形式为

str1

P	r	o	g	r	a	m	m	i	n
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

6.4.5 字符串处理函数

- ▶ strncpy的标准用法为

```
strncpy(s1, s2, sizeof(s1)-1); // n最大为s1存储空间长度减1  
s1[sizeof(s1)-1]='\0'; // 最后放上空字符结束
```

- ▶ strcpy复制字符串时可能会由于s1存储空间小导致数组越界，而strncpy可以避免。

6.4.5 字符串处理函数

- ▶ (3) 字符串连接函数strcat (string concatenate)

```
char *strcat(char *s1, const char *s2);
```

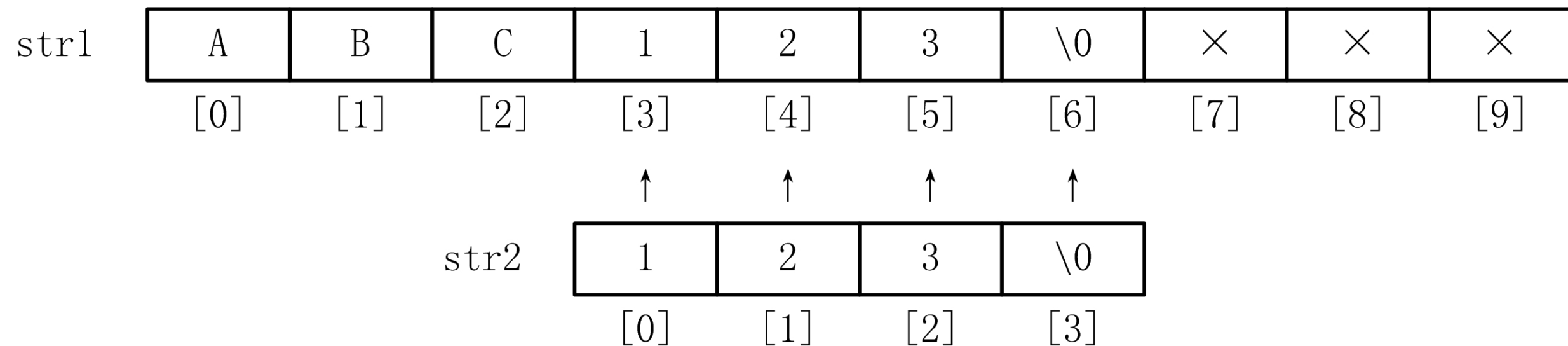
- ▶ strcat将s2字符串连接到s1的后面，包括空字符。s1是字符数组或指向字符数组的指针，其长度应该足够大，以便能容纳连接的字符串；s2可以是字符串常量、字符数组或指向字符数组的指针。

6.4.5 字符串处理函数

► 例如：

```
char str1[10]="ABC", str2[]="123";  
strcat(str1,str2); //在str1后面连接str2,str2未变化
```

► 运行过程如图所示。



6.4.5 字符串处理函数

- ▶ 可以连接字符串常量。例如

```
strcat(str1, "Java");
```

6.4.5 字符串处理函数

► (4) 字符串连接函数strncat

```
char *strncat(char *s1, const char *s2, size_t n);
```

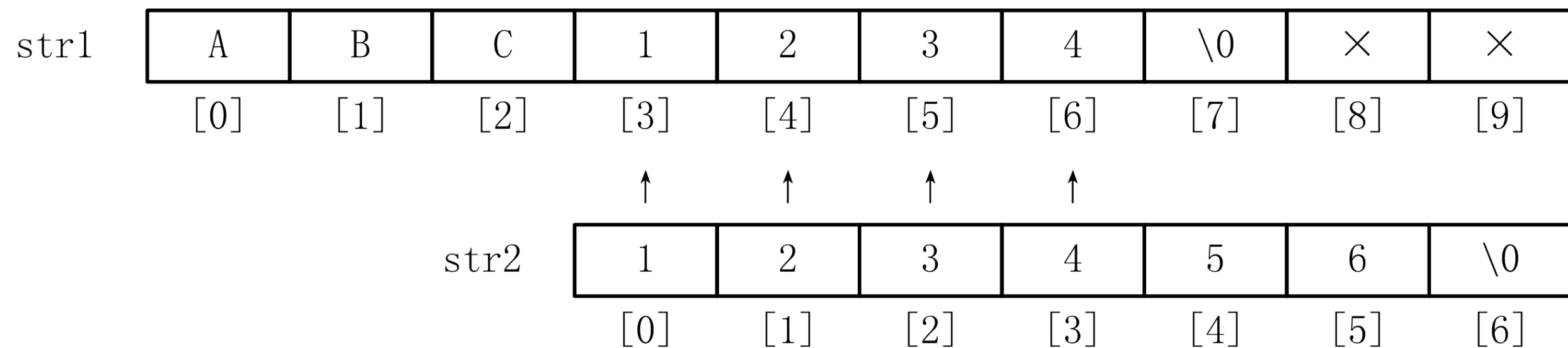
- strncat将s2中不超过n个字符的字符串连接到s1的后面，其他与strcat函数类似。

6.4.5 字符串处理函数

► 例如：

```
char str1[10]="ABC",str2[]="123456";  
strncat(str1,str2,4);
```

► 运行过程如图所示。



6.4.5 字符串处理函数

- ▶ strcat连接字符串时可能会由于s1存储空间小导致数组越界，而strncat可以避免。

6.4.5 字符串处理函数

- ▶ (5) 字符串比较函数strcmp (string compare)

```
int strcmp(const char *s1, const char *s2);
```

- ▶ strcmp比较字符串s1和s2的大小。s1和s2可以是字符串常量、字符数组或指向字符数组的指针，比较结果为：
 - ▶ ①若s1大于s2，返回大于0的整数值；
 - ▶ ②若s1等于s2，返回0；
 - ▶ ③若s1小于s2，返回小于0的整数值。

6.4.5 字符串处理函数

- ▶ 字符串比较的规则是对两个字符串自左向右依次比较字符的ASCII数值，直到出现不同的字符或遇到空字符为止。若全部字符相同，则认为字符串相等；若出现不同的字符则以第一个不相同的字符的比较结果为准。
- ▶ 例如："A"小于"B"，"A"小于"a"，"The"大于"This"，"31"大于"25"等，依次类推。一般地，数字字符小于字母、大写字母小于小写字母、英文小于汉字等。

6.4.5 字符串处理函数

- ▶ 两个字符串比较大小，不能使用关系运算符，例如：

```
if (str1>str2) ..... // 不是字符串比较的含义
```

- ▶ 而应该使用strcmp函数，例如：

```
if (strcmp(str1,str2)==0) ..... //比较字符串相等  
if (strcmp(str1,str2)!=0) ..... //比较字符串不相等  
if (strcmp(str1,str2)>0) ..... //比较str1大于str2  
if (strcmp(str1,str2)<0) ..... //比较str1小于str2
```

6.4.5 字符串处理函数

- ▶ (6) 计算字符串长度函数 (string length)

```
size_t strlen(const char *s);
```

- ▶ strlen返回字符串s的长度。s可以是字符串常量、字符数组或指向字符数组的指针。

6.4.5 字符串处理函数

► 例如：

```
char str[20]="Visual Basic";  
n=strlen("Language"); //n=8  
n=strlen(str); //n=12  
n=sizeof str; //n=20
```

► strlen计算字符串的长度， sizeof计算字符数组的长度。

6.4.5 字符串处理函数

► (7) 字符串转换成数值函数

```
#include <stdlib.h>
double atof(const char *ns); //将字符串数值转换为double数据
int atoi(const char *ns); //将字符串数值转换为int数据
```

- 两个函数可以将数值内容的字符串转换为数值类型的值，atof转换为双精度浮点型值，atoi转换为整型值。ns可以是字符串常量、字符数组或指向字符数组的指针，但内容必须是对应类型的合法数据。

6.4.5 字符串处理函数

► 例如：

```
f=atof("123.456"); //f=123.456  
i=atoi("123"); //i=123  
i=atoi("-456"); //i=-456
```

► 转换函数在解析字符串数值时，只要遇到不合法字符就结束转换，例如：

```
f=atof("12.3.456"); //f=12.3  
i=atoi("a123"); //i=0
```


6.4.5 字符串处理函数

► (8) 数据写入字符串的格式化输出函数

```
#include <stdio.h>
int sprintf(char *s, const char *format, ...);
//“输出”格式化数据到字符数组中
```

- sprintf与printf功能类似，都是输出格式化的数据，但sprintf“输出”到字符串s中。s是字符数组或指向字符数组的指针，其长度应该足够大，以便能容纳输出信息。
- sprintf输出后会在数据的后面增加空字符，使s成为字符串。

6.4.5 字符串处理函数

► 例如：

```
char str[10];  
sprintf(str, "%d*%d=%d", 2, 3, 2*3); // 输出结果不显示，存储在str中
```

► 执行后str的存储形式为：

str

2	*	3	=	6	\0	×	×	×	×
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

6.4.5 字符串处理函数

► (9) 从字符串读入数据的格式化输入函数

```
#include <stdio.h>
int sscanf(const char * s, const char * format, ...);
//从字符串中“输入”格式化数据
```

- sscanf与scanf功能类似，都是输入格式化的数据；但sscanf从字符串s中读取数据。s可以是字符串常量、字符数组或指向字符数组的指针。

6.4.5 字符串处理函数

► 例如：

```
int a,b;  
sscanf("12 34", "%d%d", &a, &b); //读入a=12 b=34
```

6.4.5 字符串处理函数

- ▶ 有了sprintf、sscanf、atof这些函数，就可以实现字符串文本信息与数值型数据相互转换，这是非常实用的功能。

6.4.5 字符串处理函数



【例6.9】

将3个字符串按由小到大的顺序输出。

6.4.5 字符串处理函数

例6.9

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char s1[10]="Java", s2[10]="CPP", s3[10]="Basic";
6     char t[100];
7     if (strcmp(s1,s2)>0) { //s1大于s2交换
8         strcpy(t,s1); strcpy(s1,s2); strcpy(s2,t);
9     }
10    if (strcmp(s1,s3)>0) { //s1大于s3交换
11        strcpy(t,s1); strcpy(s1,s3); strcpy(s3,t);
12    }
13    if (strcmp(s2,s3)>0) { //s2大于s3交换
14        strcpy(t,s2); strcpy(s2,s3); strcpy(s3,t);
15    }
```


6.4.5 字符串处理函数

例6.9

```
16    printf("%s,%s,%s\n",s1,s2,s3);  
17    return 0;  
18 }
```


6.4.5 字符串处理函数

例6.9



1 #include <stdio.h>
2 #include <string.h>

Basic,CPP,Java

1大于s3交换
s3); strcpy(s3,t);

2大于s3交换
s3); strcpy(s3,t);

CP 程序设计