中国人民大学
RENMIN UNIVERSITY OF CHINA

第9讲 指针(1)

余力

buaayuli@ruc.edu.cn

回顾一个问题 (1)

- 将两个整数x和y的值进行调换
  - 例子：x=10; y=20 → x=20; y=10
- 设计一个函数swap实现这一功能

```
void swap (int x, int y) {
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

回顾一个问题 (2)

```
void swap(int x, int y);
int main() {
    int x=10, y=20;
    printf("Before swapping: x=%d, y=%d\n", x, y);
    swap(x,y);
    printf("After swapping: x=%d, y=%d\n", x, y);
}
```

```
Before swapping: x=10, y=20
After swapping: x=10, y=20
```

指针部分内容提要

1 指针的基本概念

2 指针与数组

3 指针与字符串

4 指针与函数

5 指针与结构体

中国人民大学
RENMIN UNIVERSITY OF CHINA

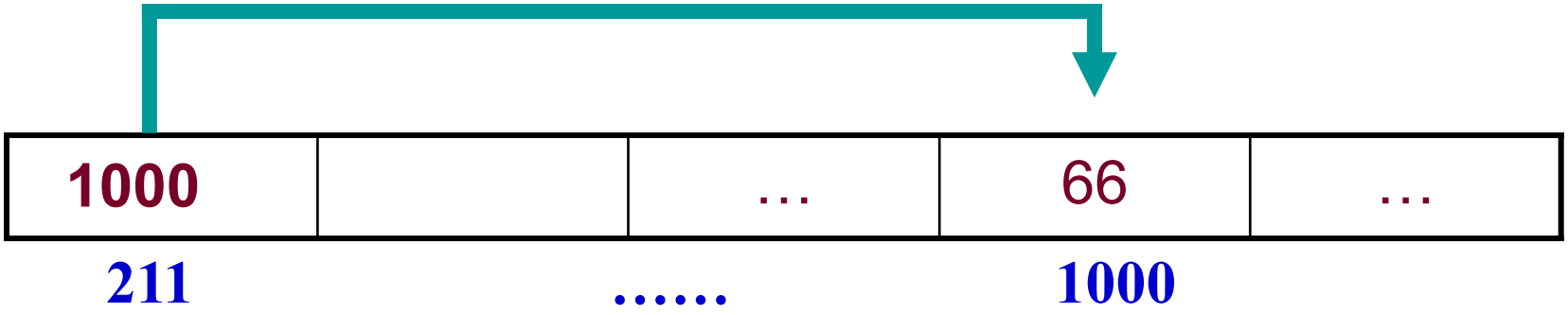
1. 指针的基本概念

什么是指针 (1)

- 指针是一类变量
  - 学过的变量：整型int、字符型char，等等
  - 指针也是一类变量，本质并无不同
- 指针的取值是内存的地址
  - 学过的变量取值
    - 整型变量的取值是一个整数，如1024
    - 字符型变量的取值是字符，如’ A’
  - 指针的取值是一个内存地址！

什么是指针 (2)

- 指针的取值是内存的地址



- 生活中的例子
  - 将信箱想象成内存(Memory)
  - 信箱221号有信息：去1000号找重要数据
  - 信箱1000号有信息：重要数据66

指针的声明 (1)

- 强调：指针只是一类特殊的变量
- 与普通变量声明的“同”
  - 变量名：这与一般变量取名相同，由英文字符开始
- 与普通变量声明的“异”
  - 指针变量的类型：是指针所指向的变量的类型，而不是自身的类型。
  - 指针的值是某个变量的内存地址。

指针的定义 (2)

- 声明的格式：

  类型标识符　*变量名;

- 示例：
  - int *p, *q;　　// 指向整数类型变量的指针
  - float *point;　　// 指向float型变量的指针
  - double *pd;　　// 指向double型变量的指针
  - char *pc;　　// 指向char型变量的指针

指针的初始化

```
int *p = NULL;
```

- 说明：
  - NULL在头文件中定义，是符号化的常量0，是唯一的一个允许赋值给指针的整数值
  - 表示指针不指向任何内存地址
  - 防止其指向任何未知的内存区域
  - 避免产生难以预料的错误发生
- 把指针初始化为NULL是好习惯

指针的赋值

- 将一个内存地址装入指针变量

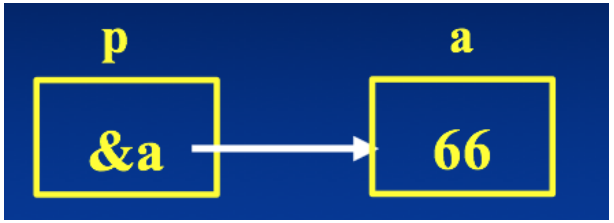
取址运算符&

- 例如：

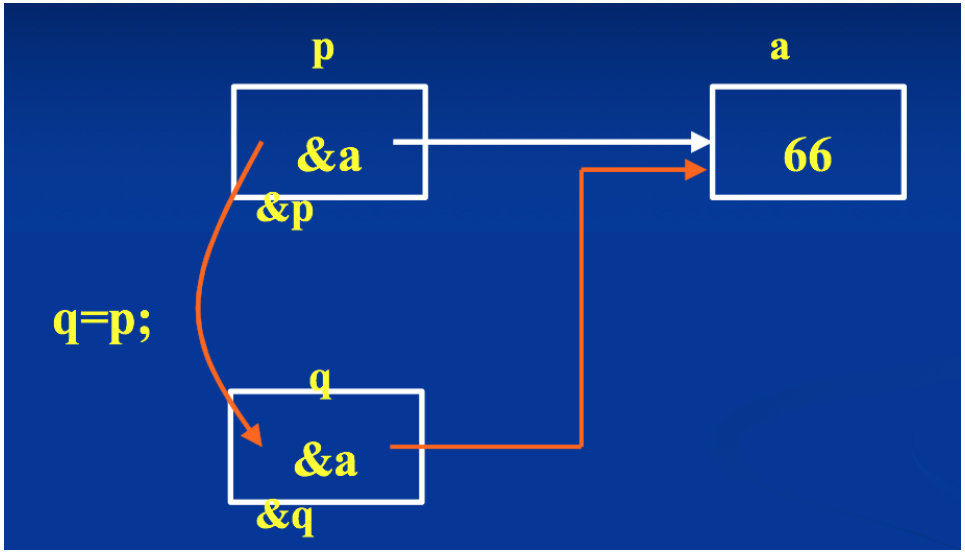
```
int a=66;                    // 定义整型变量 a, 赋初值66
// 定义p,q指针变量，赋初值为0
int *p=NULL, * q=NULL;
p = &a;                      //将变量 a 的地址赋给 p
q = p;                       // 将 p 的值赋给 q
```

画图理解指针

- p = &a;　　//将变量 a 的地址赋给 p



- q = p;

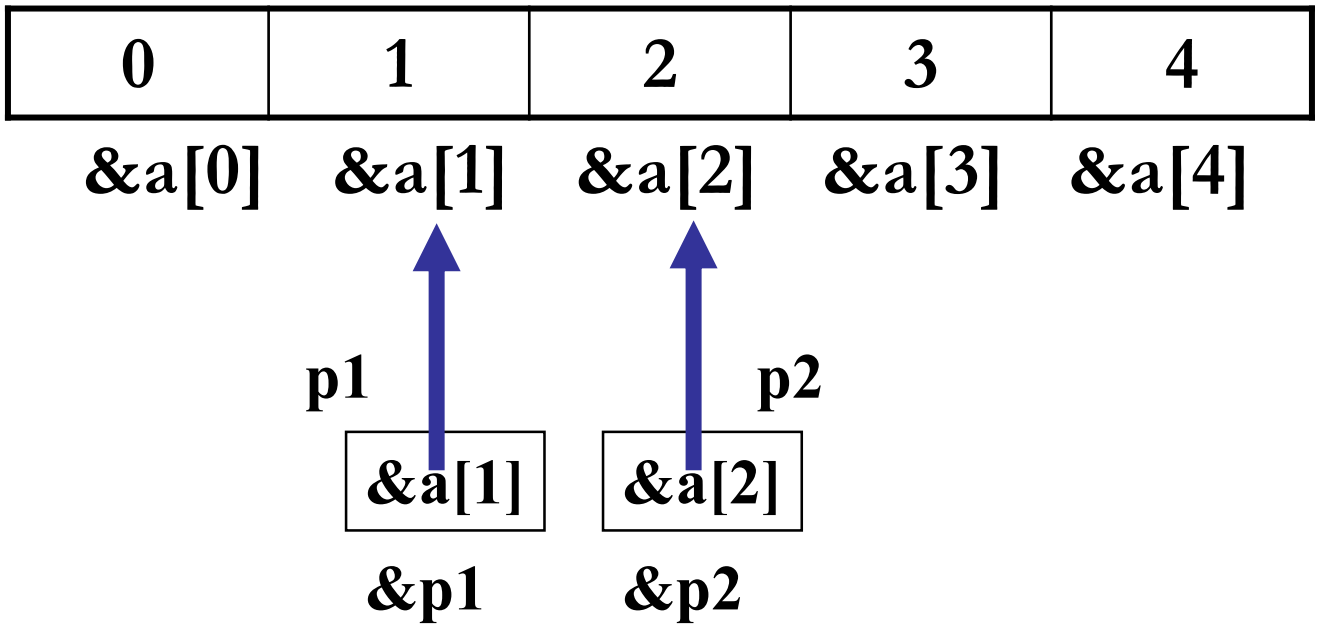


指针的赋值

```
#include <stdio.h>
int main() {
   int a[5]={0,1,2,3,4};   //定义数组，赋初值
   int *p1=NULL,*p2=NULL;  //定义指针变量
   p1=&a[1];   //赋值给指针变量，让p1指向a[1]
   p2=&a[2];   //赋值给指针变量，让p2指向a[2]
   printf("%d,%d\n", *p1, *p2);
printf("%d,%d\n", p1, p2);

   //输出a[1]和a[2]
   return 0;
```

指针的赋值



- p1 和 p2 分别指向 a[1], a[2]，这里
- & —— 取地址运算符
- * —— 指针运算符（间接访问运算符）
- *p1——间接访问p1所指向的内存单元，输出a[1]的值
- *p2——间接访问p2所指向的内存单元，输出a[2]的值

用指针实现swap函数

```
#include <stdio.h>
void interchange(int * u, int * v);
int main(void) {
  int x = 5, y = 10;
  printf("Originally x = %d and y = %d.\n", x, y);
  swap(&x,&y); /* send addresses to function */
  printf("Now x = %d and y = %d.\n", x, y);
  return 0;
}
```

```
void swap (int * u, int * v) {
  int temp;
  temp = *u;
  *u = *v;
  *v = temp;
}
```

```
void swap (int  u, int  v) {
  int temp;
  temp = u;
  u = v;
  v = temp;
}
```

```
Int* temp;
temp = u;
u = v;
v = temp;
```

行吗?

指针的指针

- 可以定义指向指针的指针
  - 二级指针，例子：int **q
  - 三级指针，例子：int ***r

```
int var = 1025;
int *p = &var;
int **q = &p;
int ***r = &q;
printf("*p = %d\n", *p);
printf("*q = %d\n", *q);
printf("**q = %d\n", **q);
printf("**r = %d\n", **r);
printf("***r = %d\n", ***r);
***r = 10;
printf（"var = %d\n", var);
**q = *p + 2;
printf（"var = %d\n", var);
```

中国人民大学
RENMIN UNIVERSITY OF CHINA

02. 指针与数组

指针 vs. 数组下标

```
#include <stdio.h>

int main() {
    int a[5]={1,3,5,7,9};
    int *p; //定义指针变量
    int i;  //定义整型变量
    p=a;//赋值给指针变量，让p指向a数组
    for(i=0; i<5; i++) {
        printf("a[%d]=%d\n", i, *p); //输出a数组元素的值
        p++;    //指针变量加1
    }
    return 0;
}
```

说明

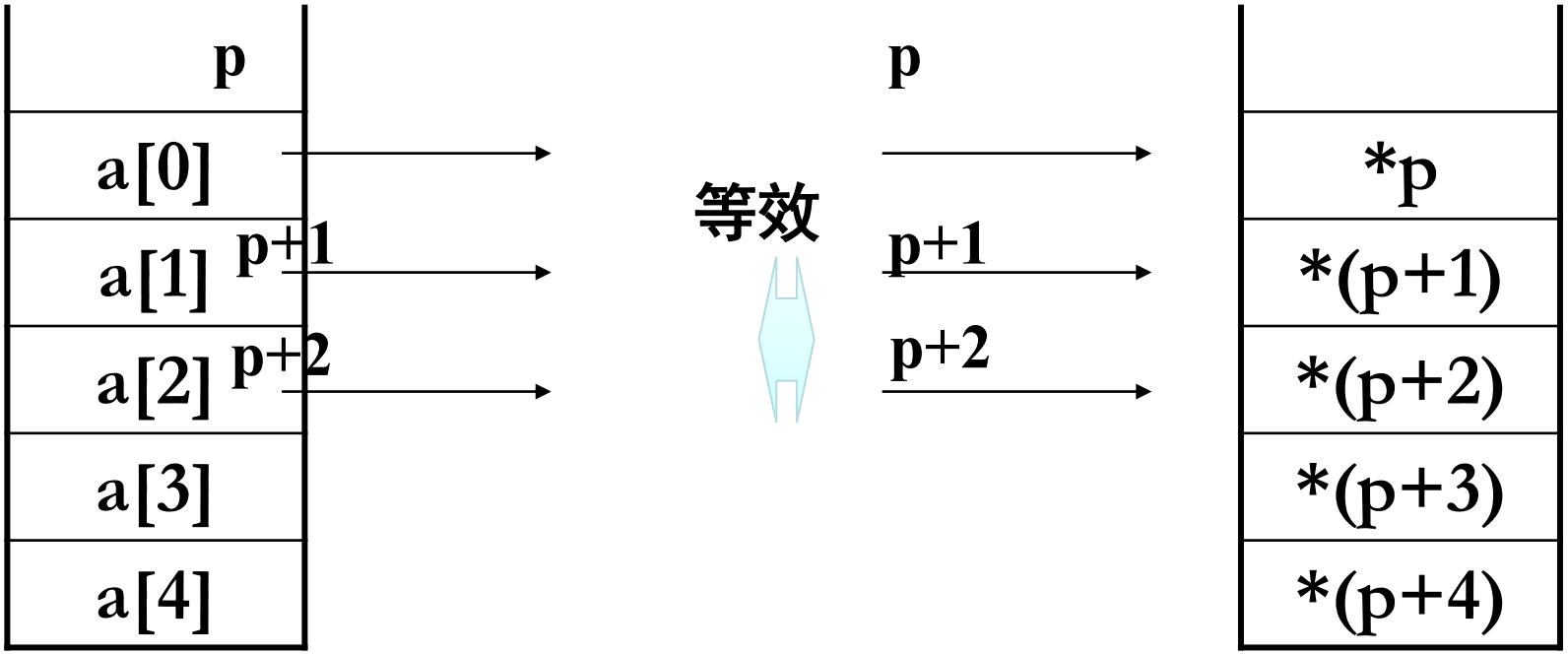
- (1) p=a; 这里数组名作为数组的起始地址，即a[0]的地址。

  因此 p=a 等效于 p=&a[0];

- (2) p=p+1; 如p指向a[0], 则p=p+1之后，p指向a[1]

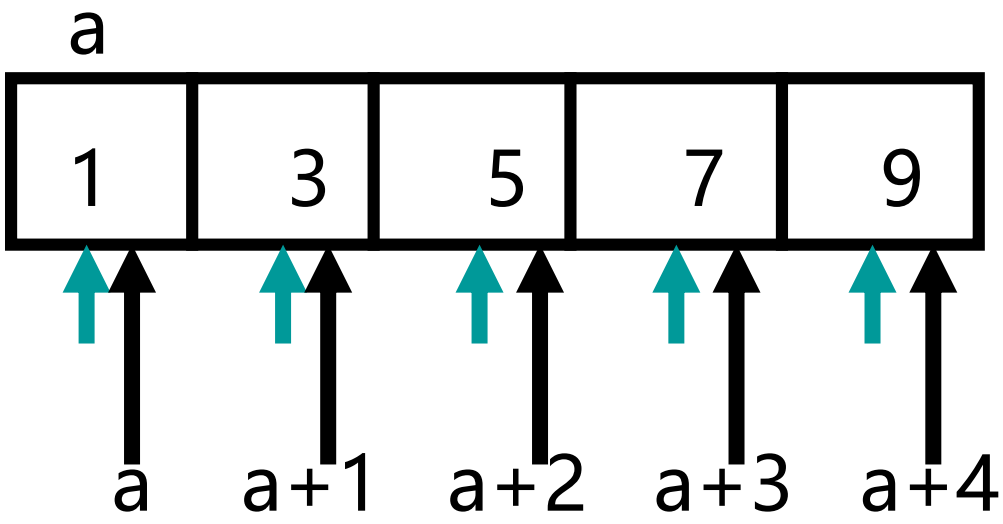
- (3) 如果p=a 等效于 p=&a[0];

  则 p=a+4 等效于 p=&a[4];



```
#include <stdio.h>          //预编译命令

int main()                  //主函数
{                           //函数体开始
  int a[5]={1,3,5,7,9};     //定义数组，赋初值
  int *p;                   //定义指针变量
  int i=0;                  //定义整型变量,赋初值
  for(p=a; p<a+5;p++)       //赋值给指针变量，让p指向a数组
  {                         //循环体开始
     printf("a[%d]=%d\n", i, *p);  //输出a数组元素的值
     i++;                   //让i加1
  }                         //循环体结束
  return 0;
}                           //函数体结束
```

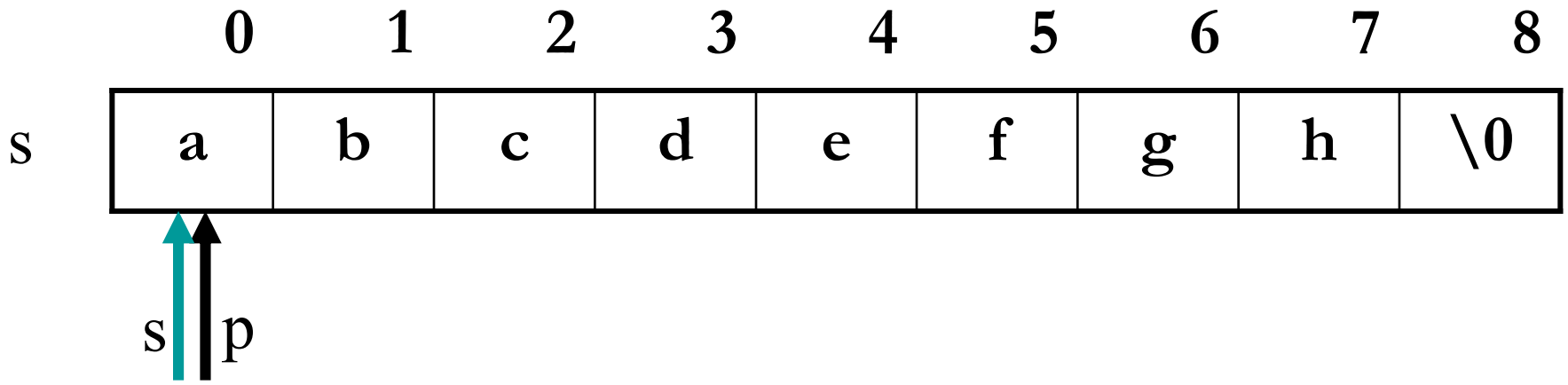


p=a p p+1 p+2 p+3 p+4

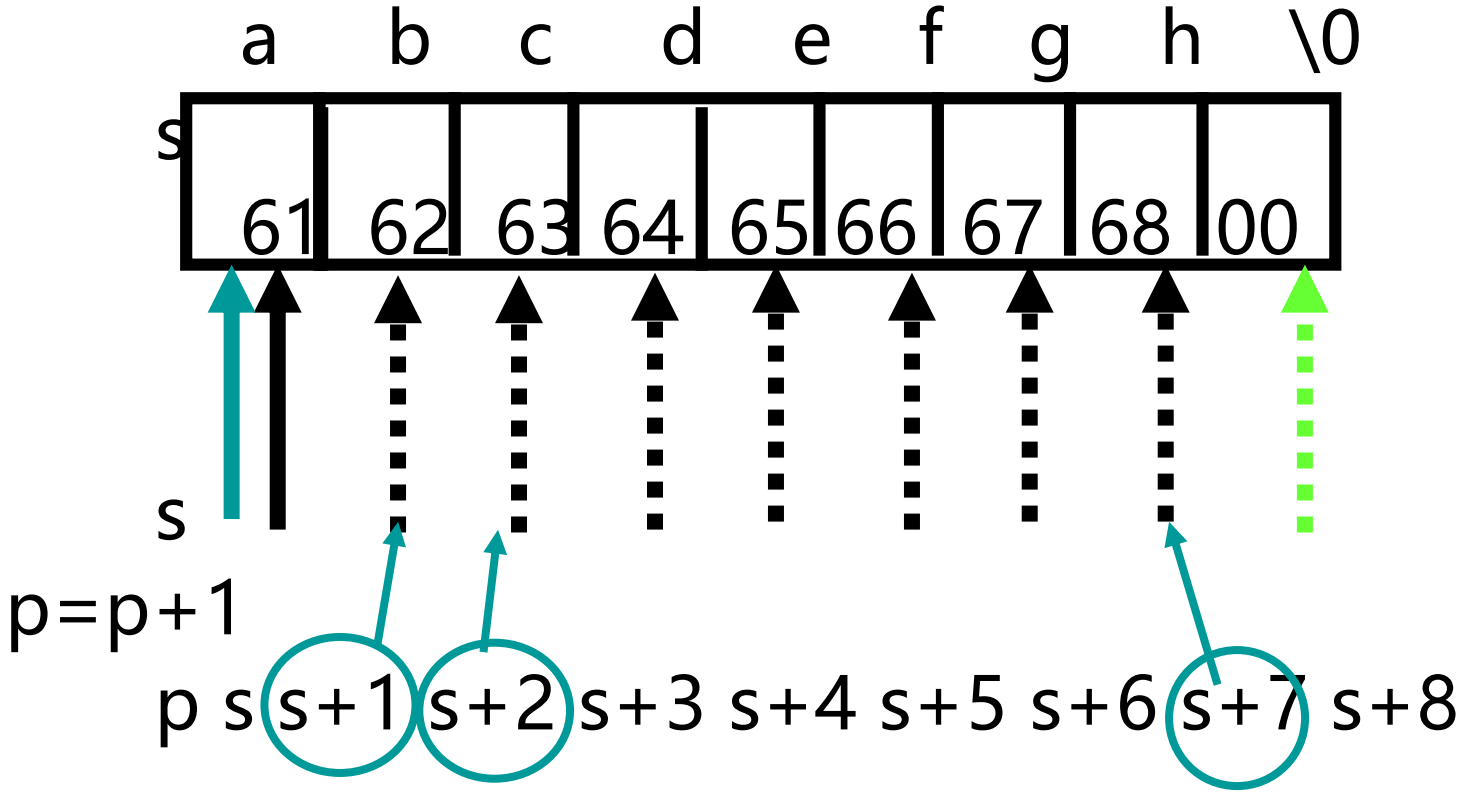
数组名是一个常量指针，指向该数组的首地址，例

```
#include <stdio.h>

int main( )
{
    char *p=NULL;                  // 定义指向字符类型的指针变量p
    char s[] = “abcdefgh”;         // 定义字符数组，并赋值
    p=s;                           // 数组名是一个常量指针，
                                   // 它指向该数组首地址
    while (*p != ‘\0’)   // 当p所指向的数组元素不为’ \0’时
    {
        p++;                       // 让指针加1
    }
    printf("字串长度为%d\n", p-s);// 输出字串长
    return 0;
}
```



图中数组的首地址是 s[0] 的地址，即&s[0]。s 可看作是指向 s[0] 的指针。s 是不会动的，是常量指针。


a b c d e f g h \0
s
61 62 63 64 65 66 67 68 00
s
p=p+1
p s s+1 s+2 s+3 s+4 s+5 s+6 s+7 s+8

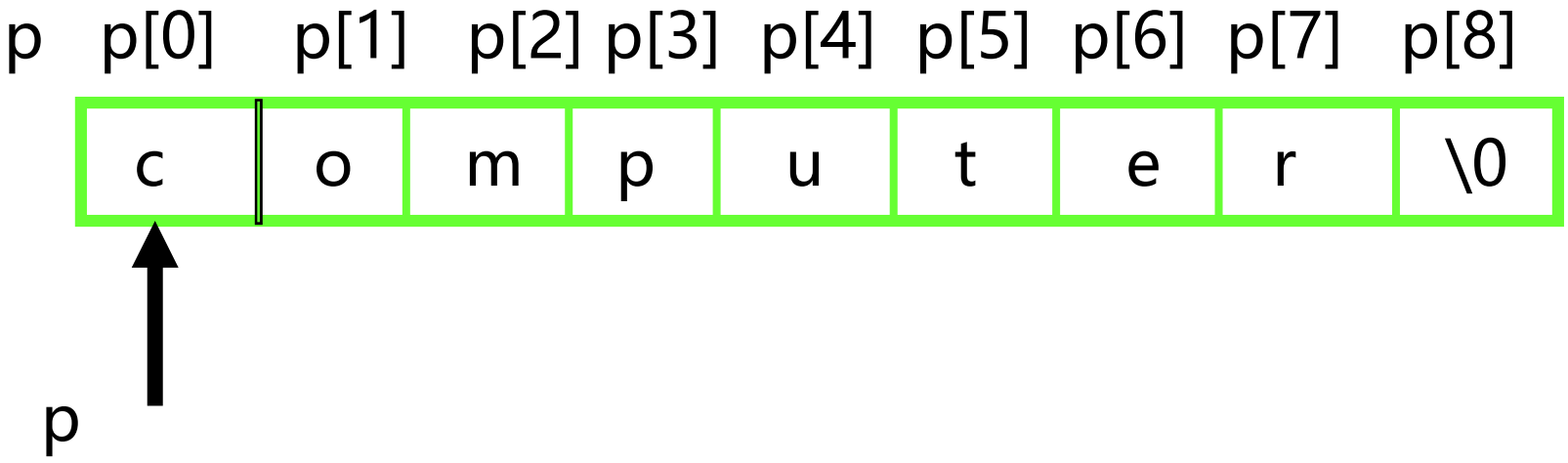
$p = s+8$ $p - s = 8$

数组名是一个常量指针，指向该数组的首地址，例

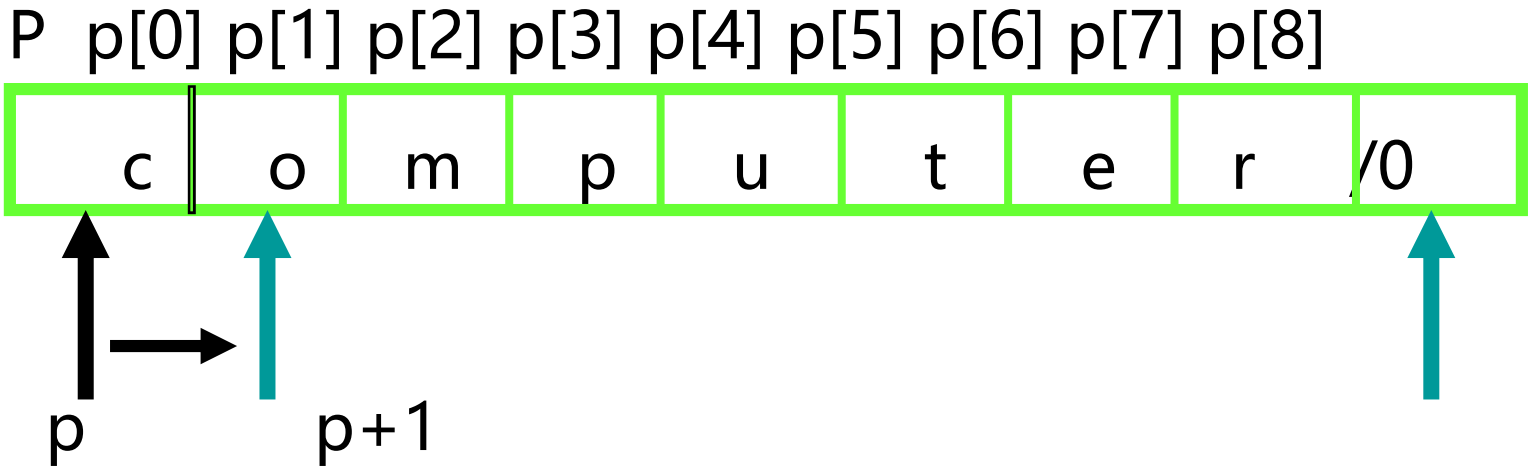
```
#include <stdio.h>                  //预编译命令
int main()                          //主函数
{                                   //函数体开始
    char shuzi[]="987654321"; //定义数组，
                                    // 赋初值为数字字符串
    char *p=&shuzi[8];              //让指针p指向shuzi[8]元素，
                                    // 该处是字符 ‘1’
    do                              // 直到型循环
    {                               // 循环体开始
        printf("%c", *p);           // 输出一个字符，该字符由p指向
        p--;                        // 让p减1
    }                               // 循环体结束
    while (p>=shuzi);               // 当p>=shuzi时，继续循环
    printf("\n");          // 换行
    return 0;
}
```

针对实现数组逆向输出

```
#include <stdio.h> //预编译命令
int main()                  //主函数
{                           //函数体开始
    int i;
    char *p;                //定义指针变量，赋初值
    p="computer";           //指针赋初值，指向字符串
    printf("%s\n", p);      //输出字符串
    for (i=0; i<8; i++)
    {                       //循环体开始
        printf("%c", p[i]);     //输出第i个字符
    }                       //循环体结束
    cout<<endl;             //换行
    while(*p)
    {                       //循环体开始
        printf("%c", *p);           //输出p所指向的字符
        p++;                //指针变量值加1
    }                       //循环体结束
    cout<<endl;             //换行
    return 0;
}                           //函数体结束v
```



数组名是常量指针，p可理解为这个字符数组的名字。



*p==c, *(p+1) ==o, *(p+2) ==m

对字符指针和字符数组的赋值

1、对字符指针变量赋值的写法

（1）char *p;
　　p = “computer”;

（2） char *p=“computer”;

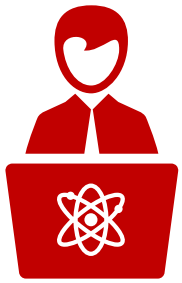
以上两种都行。可以整体赋值。

2、对字符数组赋初值的写法

（1）char as[12]=“department”;// 可以。在定义时可以整体赋值
　　char as[] = “department”;// 可以。在定义时可以整体赋值

（2）char as[12];
　　as = “department”; // 不可以！不可以整体赋值
　　as[12]=“department”; //不可以！不可以整体赋值

中國人民大學
RENMIN UNIVERSITY OF CHINA



谢谢大家！