



《计算概论A》课程 程序设计部分

指针 (2)

李 戈

北京大学 信息科学技术学院 软件研究所

2010年12月15日



北京大学



二维数组与指针

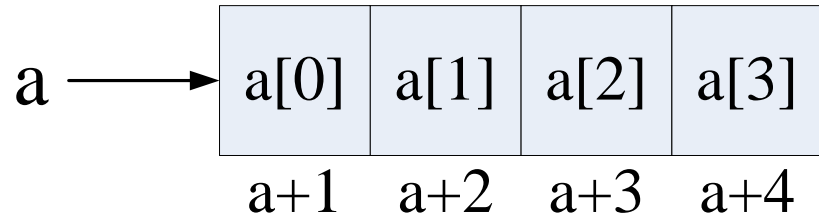


北京大学

二维数组的地址

■ 一维数组的地址

◆ `int a[4] = {1,3,5,7}`



```
#include<iostream.h>
```

```
void main()
```

```
{
```

```
    int a[4]={1,3,5,7};
```

```
    cout<<"a      = "<<a<<endl;
```

```
    cout<<"&a[0]  = "<<&a[0]<<endl;
```

```
    cout<<"a+1    = "<<a+1<<endl;
```

```
    cout<<"&a[0]+1 = "<<&a[0]+1<<endl;
```

```
    cout<<"&a[1]   = "<<&a[1]<<endl<<endl;
```

```
    cout<<"&a      = "<<&a<<endl;
```

```
    cout<<"&a+1    = "<<&a+1<<endl<<endl;
```

```
}
```

```
a          = 0x0013FF70
&a[0]      = 0x0013FF70
a+1        = 0x0013FF74
&a[0]+1    = 0x0013FF74
&a[1]      = 0x0013FF74

&a         = 0x0013FF70
&a+1       = 0x0013FF80

Press any key to continue
```

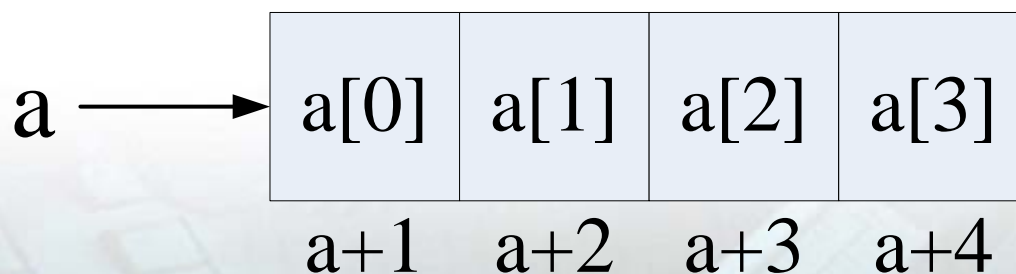


北京大学

二维数组的地址

■ 一维数组的地址

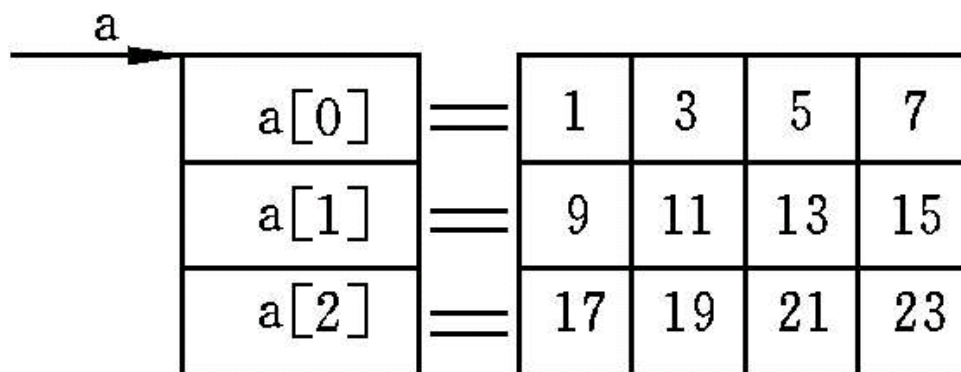
- ◆ `int a[4] = {1,3,5,7}`
- ◆ `a` 是“指向数组第一个元素”的指针；即 `a` 等价于 `&a[0]`；
- ◆ `*a` 是数组的第一个元素 `a[0]`；即 `*a` 等价于 `a[0]`；
 - `*a` 相当于“下沉”了一级；
- ◆ `&a` 是“指向数组”的指针；`&a+1` 将跨越16个字节；
 - `&a` 相当于“上浮”了一级；



二维数组的地址

■ 定义二维数组

- ◆ `int a[3][4]` 相当于 `int a[3][4]` ;
- ◆ 定义了三个存放 “`a [4]`”型数据的存储单元,
 - 他们的名字分别为: `a[0]`, `a[1]`, `a[2]`
- ◆ 二维数组**`a[3][4]`**包含三个元素: `a[0]`, `a[1]`, `a[2]`
 - 每个元素都是一个 “包含四个整型元素” 的数组





二维数组的地址

■ 二维数组的地址

◆ `int a[3][4] = {{1,3,5,7},{9,11,13,15},{17,19,21,23}};`

◆ 由对一维数组的分析可知：

“数组名是指向数组第一个元素的指针”；

◆ 且二维数组的第一个元素是`a[0]`

● `a[0]`是一个“包含四个整型元素”的一维数组；

◆ 因此可以做出各种推断：

● `a`与`&a[0]`等价；`a[0]`与`&a[0][0]`等价；

● `a[0]`与`*a`等价；`a[0][0]`与`**a`等价；



北京大學

二维数组的地址

```
#include<iostream.h>
void main()
{
    int a[3][4]={1,3,5,7},{9,11,13,15},{17,19,21,23}};

    cout<<"    a = "<<a<<endl;
    cout<<"    &a[0] = "<<&a[0]<<endl<<endl;

    cout<<"    a+1 = "<<a+1<<endl;
    cout<<"    &a[0]+1 = "<<&a[0]+1<<endl<<endl;

    cout<<"    *a = "<<*a<<endl;
    cout<<"    a[0] = "<<a[0]<<endl;
    cout<<"    &a[0][0] = "<<&a[0][0]<<endl<<endl;

    cout<<"    *a+1 = "<<*a+1<<endl;
    cout<<"    a[0]+1 = "<<a[0]+1<<endl;
    cout<<"    &a[0][0]+1 = "<<&a[0][0]+1<<endl<<endl;
}
```

	a[0]	a[0]+1	a[0]+2	a[0]+3
a	2000	2002	2004	2006
a+1	1	3	5	7
a+2	2008	2010	2012	2014
	9	11	13	15
	2016	2018	2020	2022
	17	19	21	23



二维数组的地址

```
#include<iostream.h>
```

```
void main()
```

```
{
```

```
    int a[3][4]={1,3,5,7},{9,11,13,15},{17,19,21,23}};
```

```
    cout<<"    a = "<<a<<endl;
```

```
    cout<<"    &a[0] = "<<&a[0]<<endl<<endl;
```

```
    cout<<"    a+1 = "<<a+1<<endl;
```

```
    cout<<"    &a[0]+1 = "<<&a[0]+1<<endl<<endl;
```

```
    cout<<"    *a = "<<*a<<endl;
```

```
    cout<<"    a[0] = "<<a[0]<<endl;
```

```
    cout<<"    &a[0][0] = "<<&a[0][0]<<endl<<endl;
```

```
    cout<<"    *a+1 = "<<*a+1<<endl;
```

```
    cout<<"    a[0]+1 = "<<a[0]+1<<endl;
```

```
    cout<<"    &a[0][0]+1 = "<<&a[0][0]+1<<endl;
```

```
}
```

```
    a = 0x0013FF50
```

```
    &a[0] = 0x0013FF50
```

```
    a+1 = 0x0013FF60
```

```
    &a[0]+1 = 0x0013FF60
```

```
    *a = 0x0013FF50
```

```
    a[0] = 0x0013FF50
```

```
    &a[0][0] = 0x0013FF50
```

```
    *a+1 = 0x0013FF54
```

```
    a[0]+1 = 0x0013FF54
```

```
    &a[0][0]+1 = 0x0013FF54
```

Press any key to continue.



二维数组的地址

```
#include<iostream.h>
void main()
{
    int a[3][4]={1,3,5,7},{9,11,13,15},{17,19,21,23}};

    cout<<"    a = "<<a<<endl;
    cout<<"    &a[0] = "<<&a[0]<<endl<<endl;

    cout<<"    a+1 = "<<a+1<<endl;
    cout<<"    &a[0]+1 = "<<&a[0]+1<<endl<<endl;

    cout<<"    a[1] = "<<a[1]<<endl;
    cout<<"    &a[1] = "<<&a[1]<<endl;
    cout<<"    *(a+1) = "<<*(a+1)<<endl<<endl;

    cout<<"    *a+1 = "<<*a+1<<endl<<endl;

    cout<<"    &a = "<<&a<<endl;
    cout<<"    &a+1 = "<<&a+1<<endl;
}
```

	a[0]	a[0]+1	a[0]+2	a[0]+3
a	2000	2002	2004	2006
a+1	1	3	5	7
a+2	2008	2010	2012	2014
	9	11	13	15
	2016	2018	2020	2022
	17	19	21	23



二维数组的地址

```
#include<iostream.h>
```

```
void main()
```

```
{
```

```
    int a[3][4]={1,3,5,7},{9,11,13,15},{17,19,21,23}};
```

```
    cout<<"    a = "<<a<<endl;
```

```
    cout<<"    &a[0] = "<<&a[0]<<endl<<endl;
```

```
    cout<<"    a+1 = "<<a+1<<endl;
```

```
    cout<<"    &a[0]+1 = "<<&a[0]+1<<endl<<endl;
```

```
    cout<<"    a[1] = "<<a[1]<<endl;
```

```
    cout<<"    &a[1] = "<<&a[1]<<endl;
```

```
    cout<<"    *(a+1) = "<<*(a+1)<<endl<<endl;
```

```
    cout<<"    *a+1 = "<<*a+1<<endl<<endl;
```

```
    cout<<"    &a = "<<&a<<endl;
```

```
    cout<<"    &a+1 = "<<&a+1<<endl;
```

```
}
```

```
    a = 0x0013FF50
```

```
    &a[0] = 0x0013FF50
```

```
    a+1 = 0x0013FF60
```

```
    &a[0]+1 = 0x0013FF60
```

```
    a[1] = 0x0013FF60
```

```
    &a[1] = 0x0013FF60
```

```
    *(a+1) = 0x0013FF60
```

```
    *a+1 = 0x0013FF54
```

```
    &a = 0x0013FF50
```

```
    &a+1 = 0x0013FF80
```

二维数组的地址

■ 二维数组地址

- ◆ `int a[3][4] = {{1,3,5,7},{9,11,13,15},{17,19,21,23}};`
- ◆ 数组名 `a` 是“指向数组第一个元素”的指针;
- ◆ “`*a`”等价于 `a[0]`, 相当于让 `a` 下沉了一级;
- ◆ “`&a`”表示“指向二维数组”的指针, 相当于上浮了一级;

■ 几个有用的结论

- ◆ `a`, `a[0]`, `&a[0][0]` 有相同的值;
- ◆ `a+1` 表示第1行的地址;
- ◆ `*(a+1)` 表示第1行第0列的地址;
- ◆ `*a+1` 表示第0行第1列的地址;
- ◆ `a[0]+1` 表示第0行第1列的地址;

	$a[0]$	$a[0]+1$	$a[0]+2$	$a[0]+3$
a	2000 1	2002 3	2004 5	2006 7
$a+1$	2008 9	2010 11	2012 13	2014 15
$a+2$	2016 17	2018 19	2020 21	2022 23



二维数组的元素

练习:

a

a[0],*(a+0),*a;

a+1, &a[1],

a[1], *(a+1),

a[1]+2, *(a+1)+2, &a[1][2]

***(a[1]+2),**


***(*(a+1)+2)**

a[1][2]

	$a[0]$	$a[0]+1$	$a[0]+2$	$a[0]+3$
a	2000 1	2002 3	2004 5	2006 7
$a+1$	2008 9	2010 11	2012 13	2014 15
$a+2$	2016 17	2018 19	2020 21	2022 23



北京大學



字符串与指针



北京大学



程序分析 (1)

```
#include <iostream>
using namespace std;
int main( ) {
    char h[] = "123";
    for(int i = 0; i < 10; i++)
        cin>>h[i];
    for(i = 0; i < 10; i++)
        cout<<h[i];
    cout<<endl;
    return 0;
}
```

可以运行，但危险！

它有可能占用了不允许突破的内存边界！



北京大学



程序分析 (2)

```
#include<iostream>
using namespace std;
int main(){
    char h[ ] = "Peking";
    h[0]='a'; h[1]='b';
    h[2]='4'; h[3]='7';
    h[4]='c';
    cout<< h <<endl;
    return 0;
}
```

输出： ab47cg



北京大学



程序分析 (3)

```
#include<iostream>
using namespace std;
int main()
{
    char h[]="123456";
    h = "abcdef";
    cout<<h<<endl;
    return 0;
}
```

error C2440: “=”: 无法从“const char [7]”转换为“char [7]”





程序分析 (4)

```
#include<iostream>
using namespace std;
int main()
{
    char h[]="123456789";
    cin>>h;                //键入12345678912345
    cout<<h<<endl;        //输出12345678912345
    return 0;              //但，危险！
}
```





字符串与指针



北京大学



字符串与指针

■ 指向数组的指针

◆ `int a[10]; int *p; p = a;`

■ 指向字符串的指针

◆ 指向字符串的指针变量:

◆ `char a[10]; char *p; p = a;`



北京大学

字符串指针举例

请说明一下程序完成了什么任务:

```
int main()
{
    char a[ ]= "How are you?", b[20];
    char *p1, *p2;
    for (p1 = a, p2 = b; *p1 != '\0'; p1++, p2++)
        *p2 = *p1;
    *p2= '\0';
    cout << "string a is :" << a<<endl;
    cout << "string b is :" << b<<endl;
    return 0;
}
```



字符串指针举例

```
int main( )  
{  char buffer[10] = "ABC";  
    char *pc;  
    pc = "hello";  
    cout << pc << endl;  
    pc++;  
    cout << pc << endl;  
    cout << *pc << endl;  
    pc = buffer;  
    cout << pc;  
    return 0;  
}
```

输出：

hello

ello

e

ABC



北京大學

```
#include<iostream.h>
int main()
{
    int a = 5;
    int *pa = &a;

    int b[6] = {1, 2, 3, 4, 5, 6};
    int *pb = b;

    char c[6] = {'h', 'e', 'l', 'l', 'o', '\0'};
    char *pc = c;

    cout<< a <<endl;
    cout<< pa <<endl<<endl;

    cout<< &b[0] <<endl;
    cout<< b <<endl;
    cout<< pb <<endl<<endl;

    cout<< &c[0] <<endl;
    cout<< c <<endl;
    cout<< pc <<endl;

    return 0;
}
```



```
#include<iostream.h>
int main()
{
    int a = 5;
    int *pa = &a;

    int b[6] = {1,2,3,4,5,6};
    int *pb = b;

    char c[6] = {'h','e','l','l','o','\0'};
    char *pc = c;

    cout<< a <<endl;
    cout<< pa <<endl<<endl;

    cout<< &b[0] <<endl;
    cout<< b <<endl;
    cout<< pb <<endl<<endl;

    cout<< &c[0] <<endl;
    cout<< c <<endl;
    cout<< pc <<endl;

    return 0;
}
```

```
5
0x0013FF7C

0x0013FF60
0x0013FF60
0x0013FF60

hello
hello
hello
Press any key to continue
```

```
#include<iostream.h>
int main()
{
    int a = 5;
    int *pa = &a;

    int b[6] = {1,2,3,4,5,6};
    int *pb = b;

    char c[6] = {'h','e','l','l','o','\0'};
    char *pc = c;

    cout<< a <<endl;
    cout<< pa <<endl<<endl;

    cout<< &b[0] <<endl;
    cout<< b <<endl;
    cout<< pb <<endl<<endl;

    cout<<static_cast<void*>(&c[0])<<endl;
    cout<<static_cast<void*>(c)<<endl;
    cout<<static_cast<void*>(pc)<<endl;

    return 0;
}
```

5

0x0013FF7C

0x0013FF60

0x0013FF60

0x0013FF60

0x0013FF54

0x0013FF54

0x0013FF54

Press any key to continue



字符串指针举例

■ 问题

- ◆ 将一个位数不定的整数三位分解后输出
- ◆ 如：123456789 123, 456, 789

■ 要求：

- ◆ 按照输入整数的长短，动态申请存储空间

■ 思路：

- ◆ 将整数存入字符数组，并加上逗号，然后输出
 - 需要判断字符数组的长度（整数长度+逗号个数）
 - 按照整数转换成字符的规律，从后向前赋值；



```

int main() {
    int num, ilen, clen, k = 0;
    cout<<"Please input a number ";
    cin >> num;
    ilen = getlength(num);
    clen = ilen + ilen / 3;
    char *p = new char [clen+1];
    p = p + clen;
    *p = '\0';
    while (num != 0) {
        * (--p) = num % 10 + '0';
        num = num / 10;
        k++;
        if (k == 3) {
            * --p = ',';
            k = 0;
        }
    }
    if (*p == ',')
        cout<<p+1<<endl;
    else
        cout << p << endl;
    return 0;
}

```

```

int getlength(int num) {
    int count = 0;
    while (num != 0) {
        num = num / 10;
        count ++;
    }
    return count;
}

```



运算符new与delete

- new与delete是C++操作符，用于动态分配内存空间，并返回一个指向该内存空间的指针；

```
int main( ){  
    int n;  
    cin>>n;  
    int *p = new int[n];  
    for(int i = 0; i < n; i++)  
        p[i] = i;  
    for(int i = 0; i < n; i++)  
        cout<<p[i];  
    delete[ ] p;  
    return 0;  
}
```





好好想想，有没有问题？

谢谢！



清华大学